FUZZY INFERENCE SYSTEM DEVELOPMENT FOR REFRACTORY BRICK MANUFACTURING PLANT

by
Maj Mohan Kumar (Indian Army)

Department of Electrical Engineering/ACES
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

December 2001

FUZZY INFERENCE SYSTEM DEVELOPMENT FOR REFRACTORY BRICK MANUFACTURING PLANT

A Thesis submitted
in partial fulfilment of the requirement
for the degree of

MASTER OF TECHNOLOGY

by

Maj Mohan Kumar (Indian Army)

Department of Electrical Engineering/ACES
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

December 2001

CERTIFICATE

This is to certify that the work contained in this thesis entitled "Fuzzy Inference System Development For Refractory Brick Manufacturing Plant", by Maj Mohan Kumar, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr P K Kalra
Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Dec 2001

ACKNOWLEDGEMENT

It is with great pleasure that I express my deepest gratitude to Dr PK Kalra for his guidance, co-operation and patient supervision during this programme. I am thankful to the tremendous facilities in the lab and the freedom of work given by him. Besides the academic guidance, his words always provided me with a great deal of moral support at times when things appeared to be difficult for me But for his able guidance, it would have been impossible for me to work in the field of fuzzy logic.

I also express my gratitude to Sqn Ldr P A Patil for his kind co-operation in all fields of the thesis work. I also thank Prasanth, Madhav Krishna, Major Jayesh, Major Rajesh Vaid, Saleem, Manoj and Shankar for their co-operation and good wishes during our long hours of work together.

I would also like to express my gratitude to the engineers of SAIL Dr. N. K. Ghosh, Mr. P Barua and Mr. S. K. Nandi who worked long hours into the night along with us for the fruitful completion of the project, but for their untiring efforts the project would not had any meaningful end.

ABSTRACT

The past few years have witnessed a rapid growth in the number and variety of applications of fuzzy logic. The applications range from consumer products such as cameras, camcorders, washing machines, and microwave ovens to industrial process control, medical instrumentation, decision-support systems, and portfolio selection.

The basic concept underlying fuzzy logic is that of a linguistic variable, that is, a variable whose values are words rather than numbers. Although words are inherently less precise than numbers, their use is closer to human intuition. Furthermore, computing with words exploits the tolerance for imprecision and thereby lowers the cost of solution.

Another basic concept in fuzzy logic is that of a fuzzy if-then rule or, simply, fuzzy rule. The guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low cost solution. Among various combinations of methodologies in soft computing, the one that has highest visibility is that of fuzzy logic.

Hence the fuzzy logic toolbox of Matlab has been used to develop the fuzzy inference system for manufacture of converter lining bricks for Rourkela Steel Plant. Data mining tools such as CART (Classification and regression trees) has been used for rule generation from data. Rule generation has also been done using ID3. Validation of rules by use of Bayesian network and probabilistic study of effect of various parameters to predict final output has been done. Two systems have been developed one based on data and the other using expert advice for the rule generation.

TABLE OF CONTENTS

LIST OF FIGURES

(Contd…)

List of Figures (contd.)

(Contd...)

List of Figures (contd.)

(Contd...)

List Of Figures (contd.)

(Contd…)

List Of Figures (contd.)

LIST OF TABLES

LIST OF ABBREVIATIONS

BOF	-	Basic Oxygen Furnace
MgO-C	-	Magnesia Carbon
LDBP	-	Lead Dolomite Brick Plant
SAIL	-	Steel Authority of India
RDCIS	-	Research and Development Center for Iron and Steel
RSP	-	Rourkela Steel Plant
ID3	-	Inductive Decision Tree
CART	-	Classification and Regression Trees
BD	-	Bulk Density
FIS	-	Fuzzy Inference system
BBN	-	Bayesian Belief Network
CTI	-	Combustion Chamber Temperature
GHDT	-	Grain Heater Drum Temperature
RT	-	Retention Time in Grain Heater
GT	-	Grain Temperature
MT	-	Mixer temperature
MIX	-	Mix temperature
ST	-	Storage time
ROLLS	-	Number of rolls
PMT	-	Press Mix Temperature
FP	-	Forming pressure
GBD	-	Green Bulk Density
CCS	-	Cold crushing strength
CP	-	Coked porosity
HMOR	-	Hot Modulus of rupture
SCR	-	Slag corrosion Resistance
OR	-	Oxidation Resistance
RC	-	Retained Carbon
MP	-	Metal Powder

TEMP	-	Temperature
OR	-	Oxidation Resistance (mm)

LIST OF ABBREVIATIONS

Conti…

MF	-	Membership function
RC	-	Retained Carbon (%)
MP	-	Metal Powder (%)
CHP	-	Chemical Purity (scaled in range 0-100)
VH	-	Very High
H	-	High
M	-	Medium
L	-	Low
VL	-	Very Low

UNITS OF MEASUREMENTS

The following units of measurements are used for the parameters throughout the text

CTI	-	Combustion Chamber Temperature (^0C)
GHDT	-	Grain Heater Drum Temperature (^0C)
RT	-	Retention Time (minutes)
GT	-	Grain Temperature (^0C)
MT	-	Mixer temperature (^0C)
MTM	-	Mixing time (minutes)
MIX	-	Mix temperature (^0C)
ST	-	Storage time (minutes)
ROLLS	-	Number of rolls (number)
PMT	-	Press Mix Temperature (^0C)
FP	-	Forming pressure (Kg/cm^2)
PITCH	-	Pitch quantity (%)
GRAPHITE	-	Graphite quantity (%)
GBD	-	Green Bulk Density (gms/cc)
Tempering	-	Tempering Status (^0C)
CCS	-	Cold crushing strength (Kg/cm^2)
CP	-	Coked porosity (%)
HMOR	-	Hot Modulus of rupture (Kg/cm^2)
SCR	-	Slag corrosion Resistance (%)
OR	-	Oxidation Resistance (mm)
RC	-	Retained Carbon (%)
MP	-	Metal Powder (%)
CHP	-	Chemical Purity (scaled in range 0-100)

CHAPTER 1

INTRODUCTION

Basic Oxygen Furnace (BOF), commonly known as Converter, is a vital link in a steel plant and the plant productivity depends largely on the trouble free run of the steel-making vessel. Magnesia carbon bricks (MgO-C) are used as lining for the converters, that are used for melting of the iron ore and other raw material used in the manufacture of steel. Due to the extreme temperatures at which the converters are operating (almost 2000 degree centigrade) the lining deteriorates after a specific number of manufacturing cycles.

MgO-C bricks are manufactured at Rourkela Steel Plant in a joint project with Research and Development Centre for Iron and Steel (RDCIS), Ranchi. At present SAIL engineers have been able to achieve a maximum of up to 1100 cycles of converter operation in a single campaign, after which the MgO-C brick lining has to be changed leading to a down time of 10 to 15 days. By increasing the life of the bricks the number of manufacturing cycles per campaign can be increased, there by reducing the frequency of converter lining changes, and hence increasing the productivity and reducing cost and saving on man hours. With the globalisation of the market the public sector organisation like SAIL have come under excess pressure to optimise their production process or be left behind in the competition for market share.

1.1 Problem statement

To develop a fuzzy inference system model for refractory brick manufacturing process at Rourkela steel plant.

This process involved gathering the vast knowledge and experience of the engineers, workers and plant operators. Collection of online data during the manufacturing process to substantiate the knowledge acquired and analysing the manufacturing process to determine the parameters that have been taken into consideration for manufacture of high quality bricks.

1.2 Organisation of Thesis

Since a thorough understanding of the under lying process for manufacture of the bricks is very important to design the system, chapter 2 has been devoted entirely to explain the process in detail. The parameters that critically affect the brick properties have been broadly identified. In addition the properties of the raw materials

used and the optimised norms to be followed have been tabulated, which gives a deeper insight into understanding of the system.

The software tools used to implement the system are WIN-PROLOG of Logic Programming Associates Ltd and Fuzzy Logic toolbox of Matlab. The use of fuzzy logic toolbox requires a good foundation of fuzzy logic and fuzzy inference systems the salient feature of which shall be discussed in chapter 3.

Data mining tool that has been used to analyse the data collected from the plant for generation of rules are ID3 and Cart4, which have been discussed in detail in chapter 4 and chapter 5.

Bayesian network has been used for validation of rules and to carryout a probabilistic study of the effect of variation in input parameters to predict final output, the details of which have been discussed in chapter 6.

The fuzzy inference system developed using the vast knowledge acquired from the experts working on the plant, right from the engineers to the workers and plant operators, is discussed in chapter 7. The fuzzy inference system developed using the data collected from the plant is discussed in chapter 8.

Comparison between the Matlab system and WIN-PROLOG system has been brought out in chapter 9.

CHAPTER 2

REFRACTORY BRICK MANUFACTURING PROCESS

2.1 Basic Concept, brick composition and purpose

Basic Oxygen Furnace (BOF), commonly known as Converter, is a vital link in a steel plant and the plant productivity depends largely on the trouble free run of the steel-making vessel. Magnesia carbon bricks (MgO-C) are used as lining for the converters, that are used for melting of the iron ore and other raw material used in the manufacture of steel. Due to the extreme temperatures at which the converters are operating (almost 2000 degree centigrade) the lining deteriorates after a specific number of manufacturing cycles. At present SAIL engineers have been able to achieve a maximum of up to 1100 cycles of converter operation in a single campaign, after which the lining has to be changed leading to a down time of 10 to 15 days. By increasing the life of the bricks the number of manufacturing cycles can be increased, there by reducing the frequency of converter lining changes, and hence increasing the productivity and reducing cost and saving on man hours. The purpose of developing the expert system is to optimise the manufacturing process so as to steadily achieve higher and higher converter operation life.

Magnesia carbon bricks essentially constitute of magnesia grains and natural graphite. But at room temperature no reaction take place between the two, hence in order to manufacture bricks, liquid binder (pitch or resin) is required. At room temperature this binder provides the strength. At higher temperature, carbon network is produced from this binder and provides necessary binding.

Manufacture of MgO-C bricks involves preparation of batch with optimum granulometry, mixing of these two main components in particular form with liquid binder, compaction under appropriate pressure to get optimum bulk density and subsequent heat treatment.

The critical thermo-mechanical properties of MgO-C bricks are largely controlled by graphite. Graphite is also responsible for high slag corrosion resistance. Flake structure and grain size distribution of the graphite is of decisive importance for the press-ability and compaction of the MgO-C bricks as well as oxidation resistance.

2.2 The Brick Making Process

2.2.1 Raw material

Seawater magnesia in different size fractions, specification of which are given in table2.1, along with Natural graphite, specification of which is given in table2.2. Pitch is used as liquid binder whose characteristics are given in table2.3.

2.2.2 Brick making

Seawater magnesia is first crushed into different grain sizes and filtered using screen sieves into the proper granulometry ratio as given in table2.4. This weighted quantity of magnesia grains (coarse, medium and fine) are collected in a scale car and discharged into the grain heaters, each batch weight is fixed at 1200 kgs. At start of the operation coarse and medium grains are charged into the grain heater drum and heated for 2-4 minutes. The heated grain is then discharged into the mixer where the fines along with the graphite are added. Hot mixing is done with molten pitch having temperature of about 160 degrees centigrade. The brick pressing is done by adjusting the hydraulic pressure to 1450 kg/sqcm.

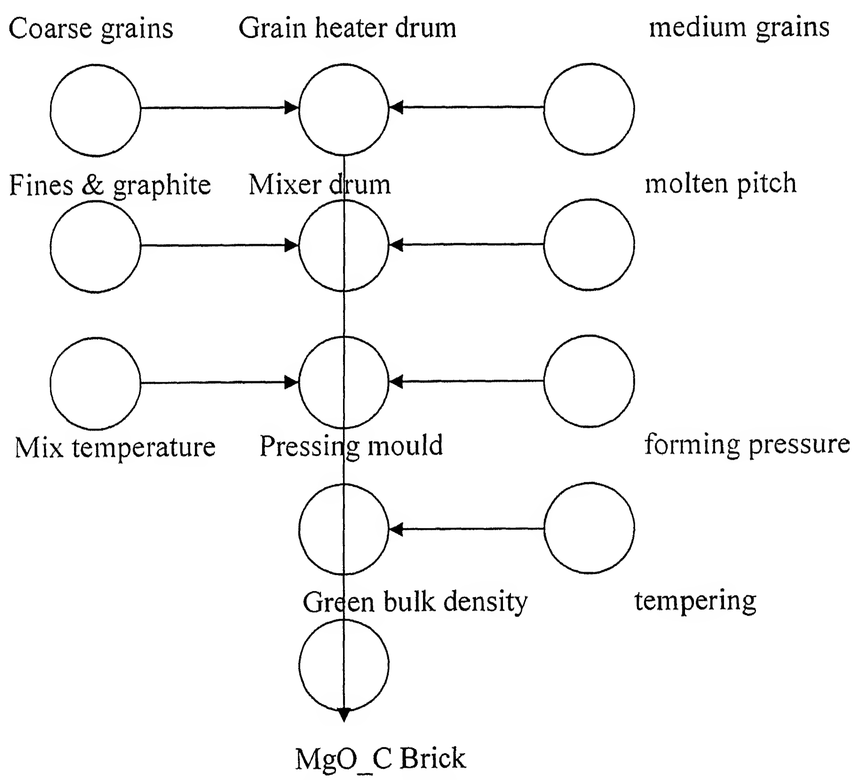


Fig 2.1 brick manufacturing process

Here the temperature of the mix before pressing is very critical as at temperatures exceeding 110 degrees centigrade the bricks start cracking, but for uniform mixing to be achieved the mixing has to be done at higher temperatures of up to 120 to 140 degrees centigrade. Thus we have two contradicting but critical requirements to be met which was done by keeping the mix in boxes for 4 to 8 hours depending on the mix temperature and then were slowly dropped from height in to the press hopper and discharged in to another box by rolling back the conveyor (which is termed as rolls). Finally the bricks are then sent for tempering and then are shrink foil packed. The tree structure of the entire process is shown in the fig 2.1.

2.2.3 Standard Optimised Operating Norms

The standardised operating norms for the various parameters that have to be strictly observed are given in table2.5.

2.3 Parameters affecting the brick properties.

The parameters that have been taken into consideration for manufacture of high quality bricks are as follows

- Granulometry
- Combustion chamber temperature
- Grain heater drum temperature
- Grain retention time
- Mixer temperature
- Mixing time
- Mix temperature
- Storage time
- Rolls
- Pitch quantity
- Graphite quantity
- Temperature of mix before pressing
- Forming pressure
- Green bulk density (GBD)

The green bulk density thus was the dependent variable, being affected by the preceding 13 independent variables, but on closer interaction with the plant engineers and the operators it was identified that the granulometry, graphite and pitch were kept constant for all campaigns and hence could be ignored. It would only be considered in case the output variation was far too high or weird. The GBD was dependent only on 10 inputs.

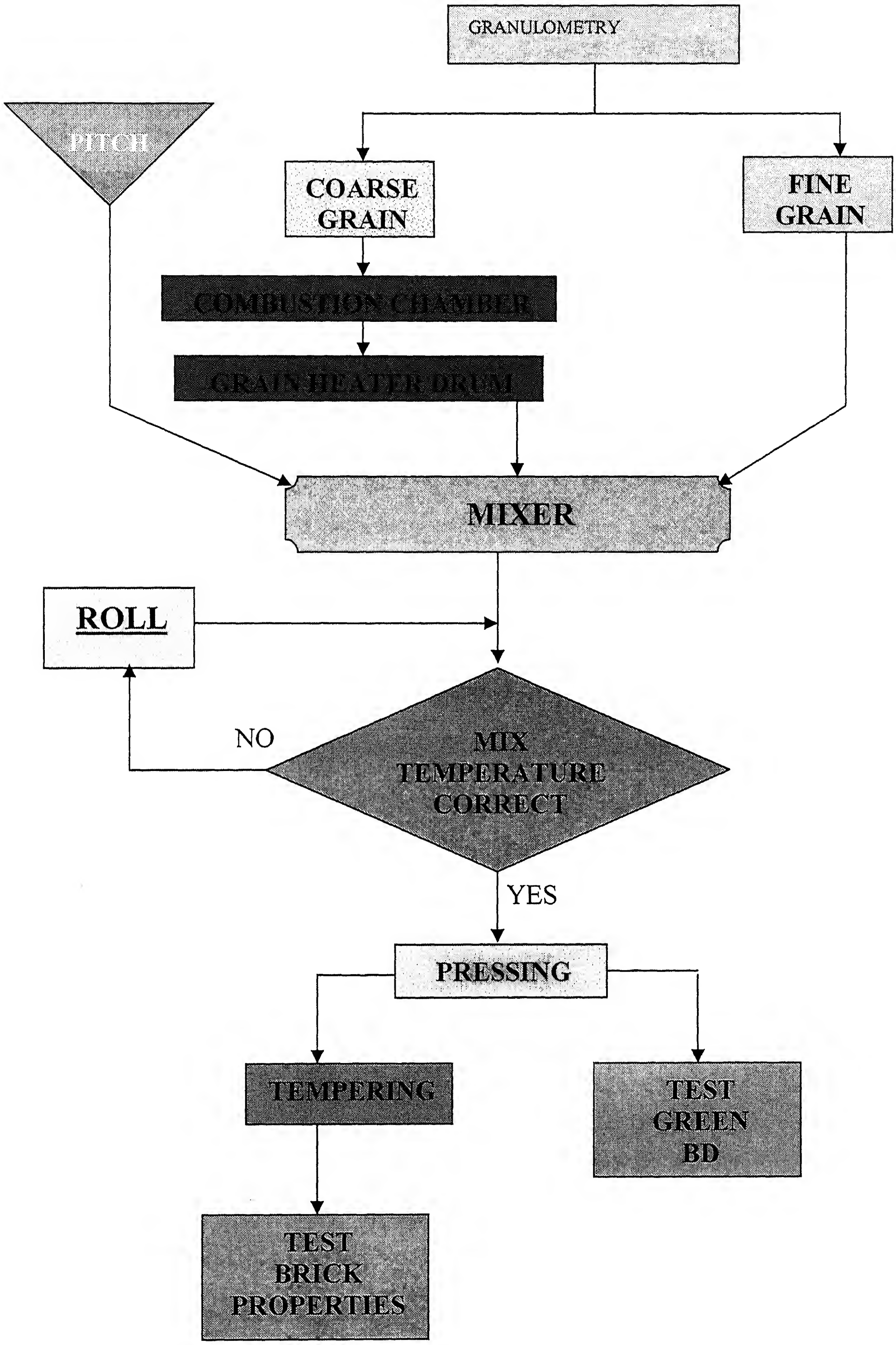


Figure 2.2: Flowchart of Brick Manufacturing Process [10]

variables, but on closer interaction with the plant engineers and the operators it was identified that the granulometry, graphite and pitch were kept constant for all campaigns

and hence could be ignored. It would only be considered in case the output variation was far too high or weird. The GBD was dependent only on 10 inputs.

The Green Bulk Density is the single most important parameter to be monitored because all other parameters in the post manufacture stage are directly or indirectly depended on it. Once the MgO_C brick has been manufactured it is sent for tempering, after which it is the bricks are warped in polypacks and stored in controlled atmosphere. The various factors affecting GBD and the parameters that are measured to check the quality of the bricks is enumerated below,

In the plant the important parameters which affect the Green Bulk Density are listed below:

- Combustion Chamber Temperature
- Grain Heater Drum Temperature
- Retention Time in Grain Heater
- Grain Temperature
- Mixer temperature
- Mixing time
- Mix temperature
- Storage time
- Number of rolls
- Press Mix Temperature
- Forming pressure
- Pitch quantity
- Graphite quantity

The important properties of the MgO-C bricks before and after tempering are listed below:

- Green Bulk Density
- Tempered Bulk Density
- Cold crushing strength
- Coked porosity
- Hot Modulus of rupture
- Slag corrosion index

2.3.1 Grain Temp (GT)

Grain temperature is the temperature of the coarse grains when being fed to the mixer. In the plant their does not exists any direct means to measure this temperature. The temperature has to be measured manually by using a thermal sensor attached to a rod by pushing it deep into the heap of grains. This temperature is given importance due to the very fact that it affects the temperature of mix and its quality directly. Also the heat transfer from coarse grains to the fine grains and powdered graphite takes place. The grain temperature depends on the following parameters:

- Combustion Chamber Temperature Initial (CTI)
- Grain Heater Drum Temperature (GHDT)
- Retention Time in Grain Heater (RT)

The more is the value of combustion chamber temperature, grain heater drum or retention time; the higher will be the grain temperature.

2.3.2 Mix Temp (MT)

Mix Temperature is the temperature of the fully prepared mixture after mixing of all the grain types, graphite and dosing of liquid pitch. If the temperature is obtained as per specification then the mix can directly taken for making of the bricks in the plant press. The Mix temperature depends on following parameters:

- Grain Temperature (GT)
- Mixer temperature (Mix_Temp)
- Mixing time

Higher the grain temperature, Mixer temperature and the Mixing Time, higher will be the mix temperature.

2.3.3 Press Mix Temperature (PMT)

Press Mix Temperature is the Mix Temperature at the time of the mix going in for making of the bricks. It happens very regularly that the Mix temperature varies from its actual required value and the mix has to be cooled down or reheated by repeating the process of heating. Cooling takes place by just leaving the drum containing mix on one side for some time and let it cool naturally. This is called Storage time. Other method to cool the mix quickly is to raise the drum to certain height and pour the mix slowly into another drum on

ground thus letting it dissipate heat to surrounding air. One such process is called rolling of the mix and this process can be repeated a number of times. Thus the Press Mix Temperature will depend on following parameters:

- Mix temperature
- Storage time
- Number of rolls

The Mix temperature is brought to the specified value of Press Mix temperature by storage time and the number of rolls. All temperatures in the plant are measured in degrees centigrade and hence for obvious reasons the units have not been mentioned explicitly any where.

2.3.4 Green Bulk Density (GBD)

This is the most important property of brick in the plant process. If the green BD is achieved correctly, the other properties given the correct composition of materials and their quality will generally will be as per the specifications. It is measured in grams per cubic centimeters and generally varies from 2.92 to 3.15. Generally the higher the density of the brick greater is the life achieved. Green BD will directly depend on the following parameters:

- Press Mix Temperature
- Forming pressure
- Pitch quantity
- Graphite quantity

2.3.5 Cold Crushing Strength (CCS)

It is the measure of the strength required to rupture the brick under compression at room temperature. It is measured in kilograms per square centimeter. Higher the GBD and Pitch higher will be the CCS, but the Tempering should to optimized, its value varies between 212 to 354 Kg/cm^2. The CCS of bricks depends on following:

- Pitch quantity
- Green Bulk Density
- Existing Tempering

2.3.6 Coked Porosity (CP)

It varies depending on the quantity of retained carbon and its value ranges from 10.82 to 15.25 %, thus it is measured as a % of the retained carbon. The CP of bricks depends on the following:

- Pitch quantity
- Green Bulk Density

2.3.7 Hot Modulus of rupture (HMOR)

Modulus of rupture is the maximum stress that a rectangular test piece of specified dimension can withstand when it is bent in a three point bending device. It is tested in both oxidizing atmosphere and reducing atmosphere, it a measure of the strength of the bricks under heated condition. The test pieces are heated to the test temperature (generally 1400°C for MgO-C). It has a higher value under reducing condition that oxidizing condition. The typical values under reducing condition vary between 60 and 105 Kg/cm^2, under oxidizing condition they vary between 15 and 35. The HMOR depends on the following:

- Green Bulk Density
- Cold crushing strength
- Carbon Content
- Aluminum powder

2.3.8 Slag Corrosion Resistance (SCR)

It generally depends on the quantity of graphite added, bricks with higher graphite showed lower SCR and those with lower graphite showed higher SCR. This is also specified in terms of Slag Corrosion index and measured as the % of wear in millimeters. Its typical values vary between 100 and 120 %. The SCR depends on the following parameters:

- Carbon Content
- Chemical purity
- Coked Porosity

Table2.1 Specification of sea water magnesia used

Properties chemical composition %	Source		
	Sardamag Italy	Periclase, Israel	Periclase, India
MgO min	97	99	97
CaO max	2	0.8	2
SiO_2 max	0.5	0.05	0.35
FeO_2 max	0.5	0.07	0.18
Al_2O_3 max	---	0.05	0.12
B_2O_3 max	0.15	0.005	0.03
Bulk density (gm/cc)	---	3.42	3.4
Crystal size (μ)	---	70-90	100

Table2.2 Typical Granulometry ratio

Grain size(mm)	Norm (%)
5-8	15
3-5	20
1.6-3	20
0.5-1.6	15
0-0.5	10
0-0.1	20

Table2.3 Typical Optimised Operating Parameter Norms

Parameters	Norms
Granulometry	Tolerance +/- 10 kg
Pitch quantity	60 +/- 5 kg
Graphite quantity	60 +/- 2 kg
Mix temperature	120 +/- 10 ^{0}C
Pressing temperature	105 +/- 5 ^{0}C
De-airing	Minimum twice
Green bulk density	$\geq$ 3.00 gm/cc
CCS (tempered)	$\geq$ 350 kg/cm^2
Bulk density (tempered)	$\geq$ 2.98 gm/cc

CHAPTER 3

FOUNDATIONS OF FUZZY LOGIC AND FUZZY INFERENCE SYSTEMS

3.1 Basic Concepts: - Fuzzy Logic

Fuzzy logic is all about the relative importance of precision: How important is it to be exactly right when a rough answer will do? Fuzzy logic is best summarised no better than in the words of the two great scientists,

So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality. —Albert Einstein

As complexity rises, precise statements lose meaning and meaningful statements lose precision. —Lotfi Zadeh

Fuzzy logic is a convenient way to map an input space to an output space. This is the starting point for everything else, and the great emphasis here is on the word "convenient." What do you mean by mapping input space to output space? Here are a few examples: You tell me how good your service was at a restaurant, and I will tell You what the tip should be. You tell me how hot you want the water, and I will adjust the faucet valve to the right setting. You tell me how far away the subject of your photograph is, and I will focus the lens for you. You tell me how fast the car is going and how hard the motor is working, and I will shift the gears for you.

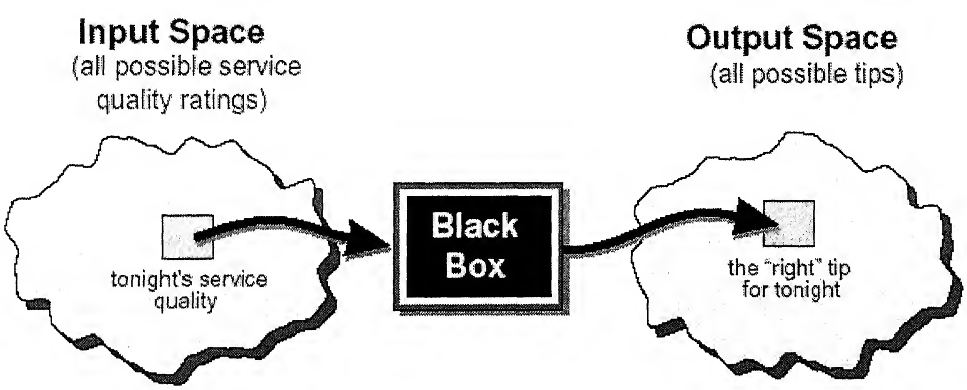


Fig 3.1 An input-output map for the tipping problem:

It's all just a matter of mapping inputs to the appropriate outputs (Fig 3.1). Between the input and the output we'll put a black box that does the work. What could go in the black box? Any number of things: fuzzy systems, linear systems, expert systems, neural

networks, differential equations, interpolated multidimensional lookup tables, or even a spiritual advisor, just to name a few of the possible options. Clearly the list could go on and on.

3.2 Why Fuzzy and not others

Of the dozens of ways to make the black box shown above work, it turns out that fuzzy is often the very best way. Why should that be? As Lotfi Zadeh, who is considered to be the father of fuzzy logic, once remarked: "In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper."

Listed below are some of the advantages offered by fuzzy logic

•Fuzzy logic is conceptually easy to understand.

The mathematical concepts behind fuzzy reasoning are very simple. What makes fuzzy nice is the "naturalness" of its approach and not its far-reaching complexity.

•Fuzzy logic is flexible.

With any given system, it's easy to massage it or layer more functionality on top of it without starting again from scratch.

•Fuzzy logic is tolerant of imprecise data.

Everything is imprecise if you look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.

•Fuzzy logic can model nonlinear functions of arbitrary complexity.

You can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like ANFIS (Adaptive Neuro-Fuzzy Inference Systems), which are available in the Fuzzy Logic Toolbox.

•Fuzzy logic can be built on top of the experience of experts.

In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets you rely on the experience of people who already understand your system.

•Fuzzy logic can be blended with conventional control techniques.

Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.

•Fuzzy logic is based on natural language.

The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.

The last statement is perhaps the most important one and deserves more discussion. Natural language, that which is used by ordinary people on a daily basis, has been shaped by thousands of years of human history to be convenient and efficient. Sentences written in ordinary language represent a triumph of efficient communication. We are generally unaware of this because ordinary language is, of course, something we use every day. Since fuzzy logic is built atop the structures of qualitative description used in everyday language, fuzzy logic is easy to use.

3.3 When Not to Use Fuzzy Logic

Fuzzy logic is not a cure-all. When should you not use fuzzy logic? The safest statement is the first one made in this introduction: fuzzy logic is a convenient way to map an input space to an output space. If you find it's not convenient, try something else. If a simpler solution already exists, use it. Fuzzy logic is the codification of common sense; use common sense when you implement it and you will probably make the right decision. Many controllers, for example, do a fine job without using fuzzy logic. However, if you take the time to become familiar with fuzzy logic, you'll see it can be a very powerful tool for dealing quickly and efficiently with imprecision and nonlinearity.

3.4 Foundations of Fuzzy Logic

Everything is vague to a degree you do not realize till you have tried to make it precise. - Bertrand Russell

3.4.1 Fuzzy Sets

Fuzzy logic starts with the concept of a fuzzy set. A *fuzzy set* is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. To understand what a fuzzy set is, first consider what is meant by what we might call a *classical set*. A classical set is a container that wholly includes or wholly excludes any given element. For example, the set of days of the week unquestionably includes Monday, Thursday, and Saturday. It just as unquestionably excludes butter, liberty, and dorsal fins, and so on. We call this set a classical set because it's been around for such a long time. It was Aristotle who first formulated the Law of the Excluded Middle, which says X must either be in set A or in set not A. Another version runs like this. Of any subject, one thing must be either asserted or denied.

Here is a restatement of the law with annotations: "Of any subject (say Monday), one thing (being a day of the week) must be either asserted or denied (I

assert that Monday is a day of the week)." This law demands that opposites, the two categories A and not-A, should between them contain the entire universe. Everything falls into either one group or the other. There is no thing that is both a day of the week and not a day of the week.

Now consider the set of days comprising a weekend. The fig 3.2 below is one attempt at classifying the weekend days.

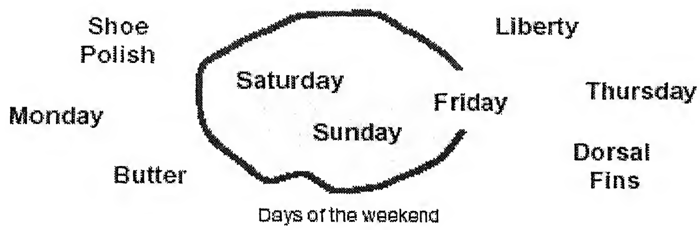


Fig 3.2 Non-classical set

Most would agree that Saturday and Sunday belong, but what about Friday? It "feels" like a part of the weekend, but somehow it seems like it should be technically excluded. So in the diagram above Friday tries its best to sit on the fence. Classical or "normal" sets wouldn't tolerate this kind of thing. Either you're in or you're out. Human experience suggests something different, though: fence sitting is a part of life.

Of course we're on tricky ground here, because we're starting to take individual perceptions and cultural background into account when we define what constitutes the weekend. But this is exactly the point. Even the dictionary is imprecise, defining the weekend as "the period from Friday night or Saturday to Monday morning." We're entering the realm where sharp edged yes-no logic stops being helpful. Fuzzy reasoning becomes valuable exactly when we're talking about how people really perceive the concept "weekend" as opposed to a simple-minded classification useful for accounting purposes only. More than anything else, the following statement lays the foundations for fuzzy logic.

In fuzzy logic, the truth of any statement becomes a matter of degree.

Any statement can be fuzzy. The tool that fuzzy reasoning gives is the ability to reply to a yes-no question with a not-quite-yes-or-no answer. This is the kind of thing that humans do all the time (think how rarely you get a straight answer to a seemingly simple question) but it's a rather new trick for computers.

How does it work? Reasoning in fuzzy logic is just a matter of generalizing the familiar yes-no (Boolean) logic. If we give "true" the numerical value of 1 and "false" the numerical value of 0, we're saying that fuzzy logic also permits in-between values like 0.2 and 0.7453. For instance,

Q: Is Saturday a weekend day?

A: 1 (yes, or true)

Q: Is Tuesday a weekend day?

A: 0 (no, or false)

Q: Is Friday a weekend day?

A: 0.8 (for the most part yes, but not completely)

Q: Is Sunday a weekend day?

A: 0.95 (yes, but not quite as much as Saturday).

Below in fig 3.5 is a plot that shows the truth-values for "weekend-ness" if we are forced to respond with an absolute yes or no response. Fig 3.3 below is a plot that shows the truth-value for weekend-ness if we are allowed to respond with fuzzy in-between values.

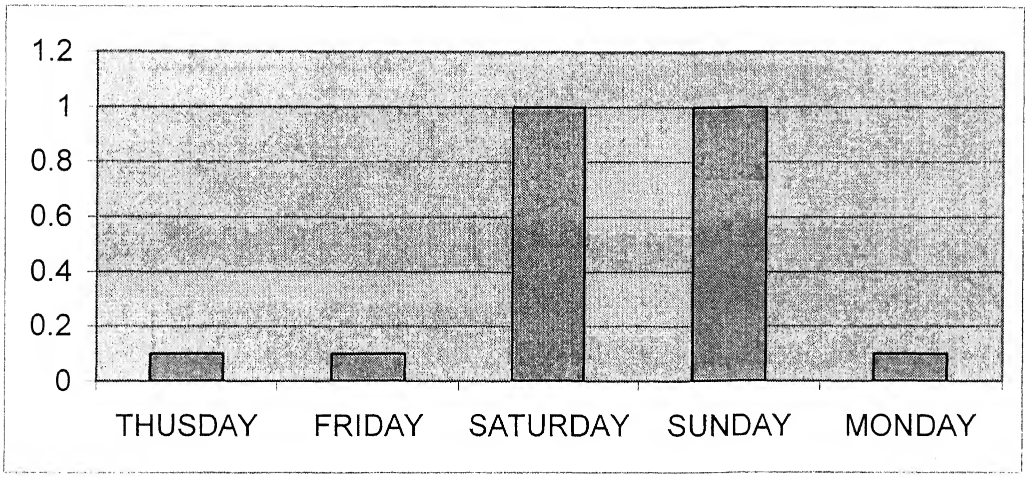


FIG 3.3 DAYS OF WEEKED TWO VALUED FUNCTION

Technically, the representation is from the domain of *multivalued logic* (or multivalent logic). If I ask the question "Is X a member of set A?" the answer might be yes, no, or any one of a thousand intermediate values in between as in fig 3.4. In other words, X might have partial membership in A. Multivalued logic stands in direct contrast to the more familiar concept of two-valued (or bivalent yes-no) logic. To return to our example, now consider a continuous scale time plot of weekend-ness shown in fig 3.6.

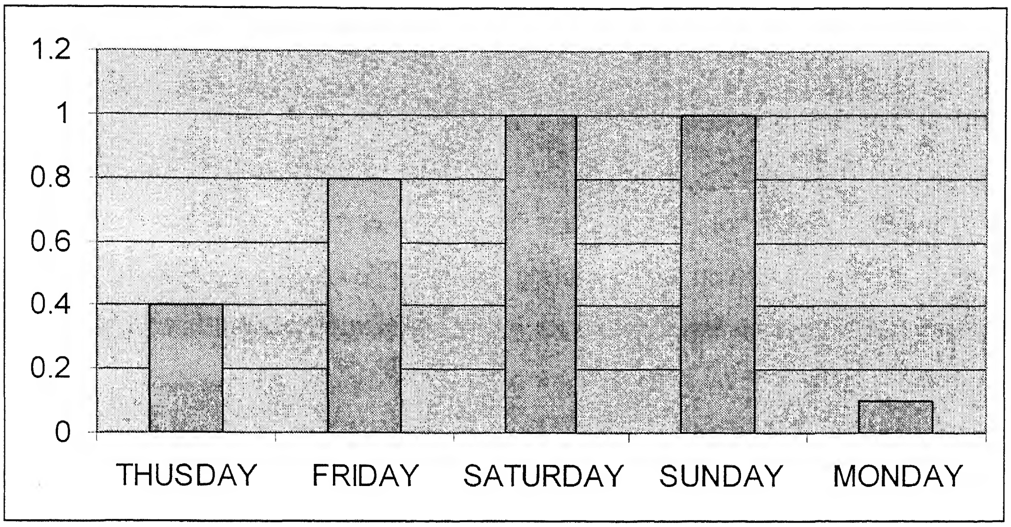


FIG 3.4 DAYS OF WEEKED MULTI-VALUED FUNCTION

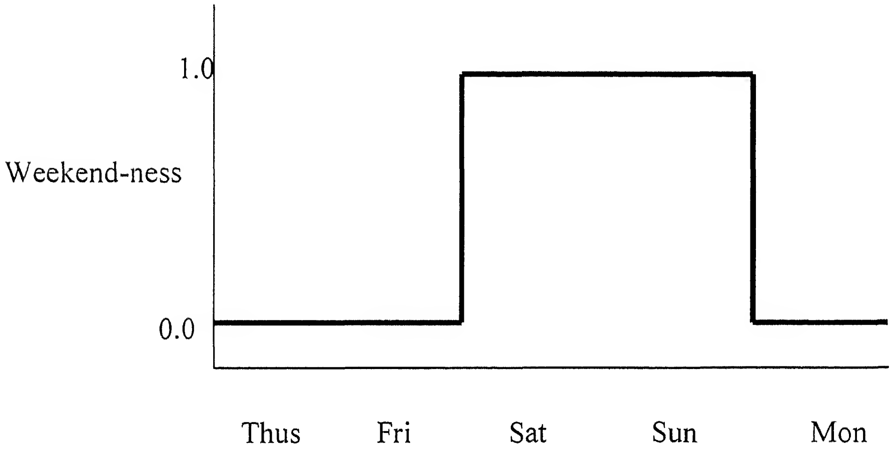


Fig 3.5 Days Of The Weekend Two – Valued Membership

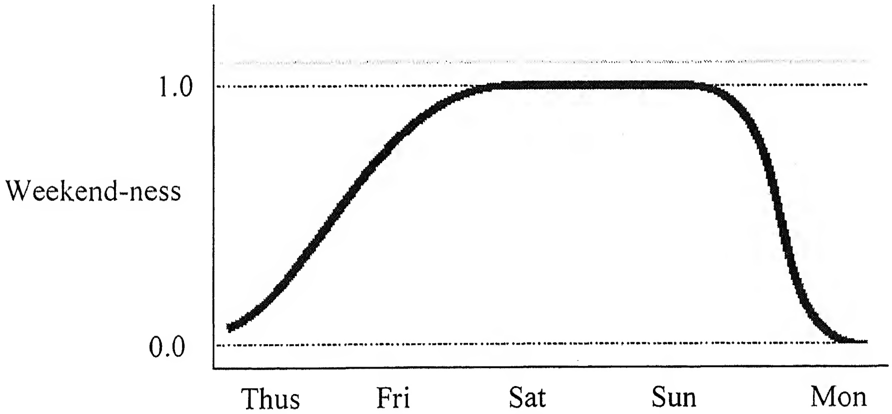


Fig 3.6 Days Of The Weekend Multi– Valued Membership

By making the plot continuous, we're defining the degree to which any given instant belongs in the weekend rather than an entire day. In the plot on the left, notice that

at midnight on Friday, just as the second hand sweeps past 12, the weekend-ness truth-value jumps discontinuously from 0 to 1. This is one way to define the weekend, and while it may be useful to an accountant, it doesn't really connect with our real-world experience of weekend-ness.

One plot shows a smoothly varying curve that accounts for the fact that all of Friday, and, to a small degree, parts of Thursday, partake of the quality of weekend-ness and thus deserve partial membership in the fuzzy set of weekend moments. The curve that defines the weekend-ness of any instant in time is a function that maps the input space (time of the week) to the output space (weekend-ness). Specifically it is known as a *membership function*.

3.4.2 Membership Functions

A *membership function* (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the *universe of discourse*.

One of the most commonly used examples of a fuzzy set is the set of tall people. In this case the universe of discourse is all potential heights, say from 3 feet to 9 feet, and the word "tall" would correspond to a curve that defines the degree to which any person is tall. If the set of tall people is given the well-defined (crisp) boundary of a classical set, we might say all people taller than six feet are officially considered tall. But such a distinction is clearly absurd. It may make sense to consider the set of all real numbers greater than six because numbers belong on an abstract plane, but when we want to talk about real people, it is unreasonable to call one person short and another one tall when they differ in height by the width of a hair.

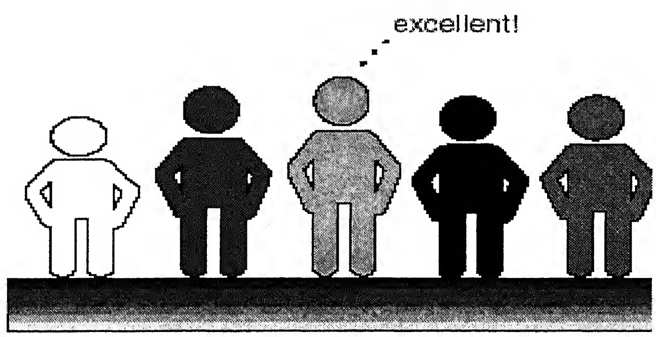


Fig 3.7 Distinction between Tall and Short

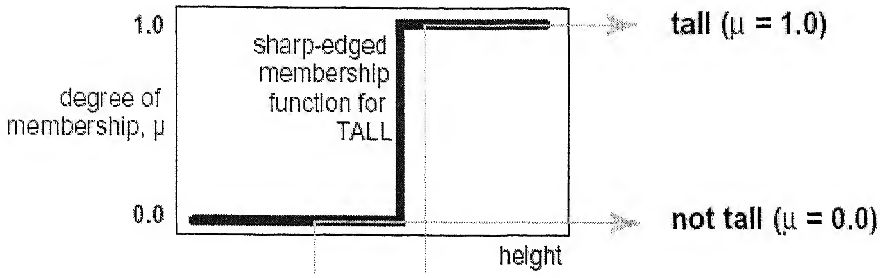


Fig 3.8 Two valued membership function

But if the kind of distinction shown in Fig 3.7 and Fig 3.8 above is unworkable, then what is the right way to define the set of tall people? Much as with our plot of weekend days, the Fig 3.9 below shows a smoothly varying curve that passes from not tall to tall. The output-axis is a number known as the membership value between 0 and 1. The curve is known as a membership function and is often given the designation of μ. This curve defines the transition from not tall to tall. Both people are tall to some degree, but one is significantly less tall than the other.

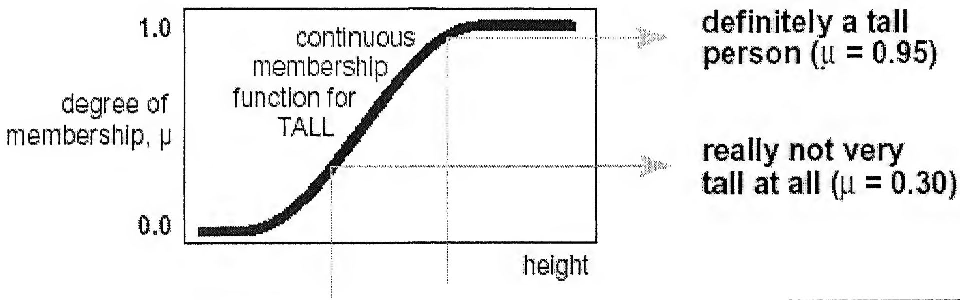


Fig 3.9 Days Of The Weekend Multi– Valued Membership

Subjective interpretations and appropriate units are built right into fuzzy sets. If I say, "She's tall," the membership function "tall" should already take into account whether I'm referring to a six-year-old or a grown woman. Similarly, the units are included in the curve. Certainly it makes no sense to say "Is she tall in inches or in meters?"

3.4.3 Logical Operations

We now know what's fuzzy about fuzzy logic, but what about the logic? The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic. In other words, if we keep the fuzzy values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. As an example, consider the standard truth tables in Fig 3.10 below.

extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. As an example, consider the standard truth tables in Table 3.1 below.

AND			OR			NOT	
A	B	A and B	A	B	A or B	A	not A
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Table 3.1 Standard Truth Tables

Now remembering that in fuzzy logic the truth of any statement is a matter of degree, how will these truth tables be altered? The input values can be real numbers between 0 and 1. What function will preserve the results of the AND truth table (for example) and also extend to all real numbers between 0 and 1? One answer is the *min* operation. That is, resolve the statement *A* AND *B*, where *A* and *B* are limited to the range (0,1), by using the function *min(A,B)*. Using the same reasoning, we can replace the OR operation with the *max* function, so that *A* OR *B* becomes equivalent to *max* (*A, B*). Finally, the operation NOT *A* becomes equivalent to the operation. Notice how the truth table above is completely unchanged by this substitution in Table 3.2 below.

Min			Max			Compliment	
A	B	Min(A,B)	A	B	Max(A,B)	A	1-A
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Table 3.2 Standard Truth Tables equivalent to min max and compliment

Moreover, since there is a function behind the truth table rather than just the truth table itself, we can now consider values other than 1 and 0.Fig 3.12 uses a graph to show the same information. We have converted the truth table to a plot of two fuzzy sets applied together to create one fuzzy set. The upper part of the figure displays plots corresponding to the two-valued truth tables above, while the lower part of the figure displays how the operations work over a continuously varying range of truth values *A* and *B* according to the fuzzy operations we have defined.

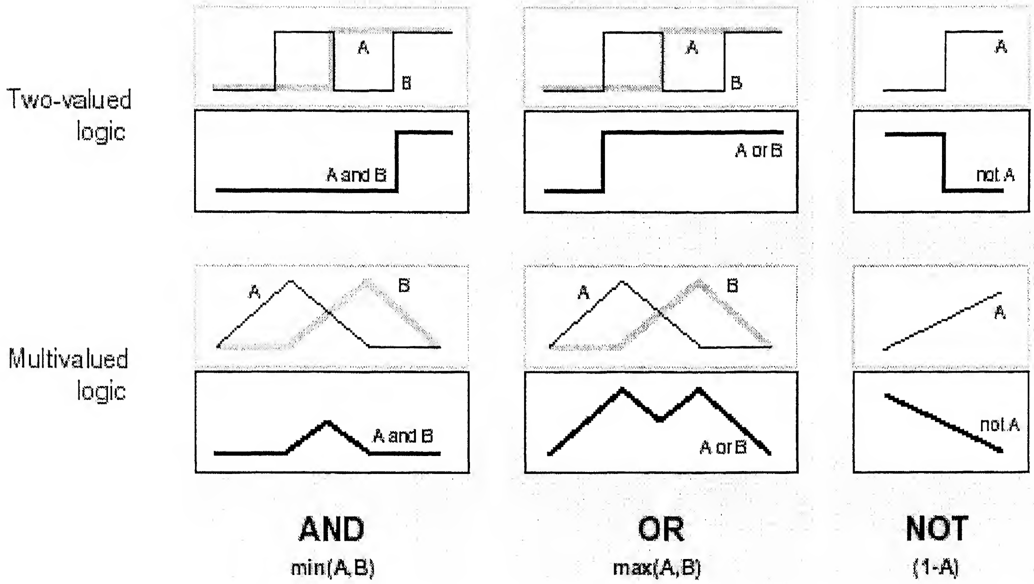


Fig 3.10 Graphical representation of Standard Operators

Given these three functions, we can resolve any construction using fuzzy sets and the fuzzy logical operation AND, OR, and NOT.

3.4.4 If-Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic [9]. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.

A single fuzzy if-then rule assumes the form

if x is A then y is B

Where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule "x is A" is called the *antecedent* or premise, while the then-part of the rule "y is B" is called the *consequent* or conclusion. An example of such a rule might be

if service is good then tip is average

Note that *good* is represented as a number between 0 and 1, and so the antecedent is an interpretation that returns a single number between 0 and 1. On the other hand, *average* is represented as a fuzzy set, and so the consequent is an assignment that assigns the entire fuzzy set B to the output variable y. In the if-then rule, the word "is" gets used in two entirely different ways depending on whether it appears in the antecedent or the consequent. A less confusing way of writing the rule would be

if service == good then tip = average

In general, the input to an if-then rule is the current value for the input variable (in this case, *service*) and the output is an entire fuzzy set (in this case, *average*). This set will later be *defuzzified,* assigning one value to the output.

Interpreting an if-then rule involves distinct parts: first evaluating the antecedent (which involves *fuzzifying* the input and applying any necessary *fuzzy operators*) and second applying that result to the consequent (known as *implication*). In the case of two-valued or binary logic, if-then rules don't present much difficulty. If the premise is true, then the conclusion is true. If we relax the restrictions of two-valued logic and let the antecedent be a fuzzy statement, how does this reflect on the conclusion? The answer is a simple one. If the antecedent is true to some degree of membership, then the consequent is also true to that same degree. In other words

in binary logic: $p \rightarrow q$ (p and q are either both true or both false)

in fuzzy logic: $0.5p \rightarrow 0.5\,q$ (partial antecedents provide partial implication)

The antecedent of a rule can have multiple parts.

if sky is gray and wind is strong and barometer is falling, then...

in which case all parts of the antecedent are calculated simultaneously and resolved to a single number using the logical operators described in the preceding section. The consequent of a rule can also have multiple parts.

if temperature is cold then hot water valve is open and cold-water valve is shut

in which case all consequents are affected equally by the result of the antecedent. How does the antecedent affect the consequent? The consequent specifies a fuzzy set be assigned to the output. The *implication function* then modifies that fuzzy set to the degree specified by the antecedent. The most common ways to modify the output fuzzy set are truncation using the *min* function (where the fuzzy set is "chopped off" as shown below) or scaling using the *prod* function (where the output fuzzy set is "squashed").

3.4.4.1 Summary of If-Then Rules

Interpreting if-then rules is a three-part process. Fig 3.13 below illustrates these parts.

(1) *Fuzzify inputs*: Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1. If there is only one part to the antecedent, this is the degree of support for the rule.

(2) *Apply fuzzy operator to multiple part antecedents*: If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule.

(3) *Apply \implication method*: Use the degree of support for the entire rule to shape the output fuzzy set. The consequent of a fuzzy rule assigns an entire fuzzy set to the output. This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent. If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

In general, one rule by itself doesn't do much good. What are needed are two or more rules that can play off one another. The output of each rule is a fuzzy set. The output fuzzy sets for each rule are then *aggregated* into a single output fuzzy set. Finally the resulting set is *defuzzified*, or resolved to a single number. The next section shows how the whole process works from beginning to end for a particular type of fuzzy inference system called a Mamdani type.

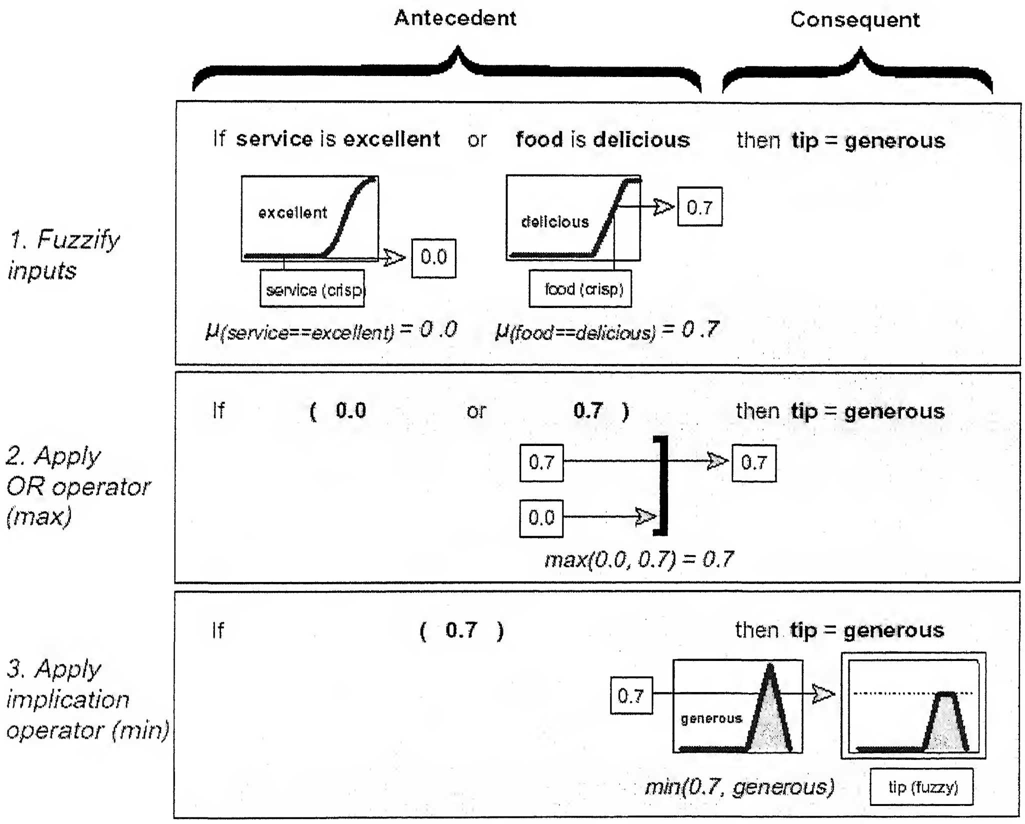


Fig 3.13 Interpreting If-Then Rules 3 part process

3.5 Fuzzy Inference Systems

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can

be made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in the previous sections: membership functions, fuzzy logic operators, and if-then rules. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined.

Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems. Since the terms used to describe the various parts of the fuzzy inference process are far from standard, we will try to be as clear as possible about the different terms introduced in this section.

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani's effort was based on Lotfi Zadeh's 1973 paper on fuzzy algorithms for complex systems and decision processes. Although the inference process we describe in the next few sections differs somewhat from the methods described in the original paper, the basic idea is much the same. Mamdani-type inference expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

It's possible, and in many cases much more efficient, to use a single spike as the output membership function rather than a distributed fuzzy set. This is sometimes known as a *singleton* output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, we use the weighted average of a few data points. Sugeno-type systems support this type of model. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant.

3.5.1 Five Steps Leading to Development of Fuzzy Inference System

Step 1. Fuzzify Inputs

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In the Fuzzy Logic Toolbox, the input is always a crisp numerical value limited to the universe of discourse of the input variable (in this case the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation.

The example we are using in this section is built on three rules, and each of the rules depends on resolving the inputs into a number of different fuzzy linguistic sets: service is poor, service is good, food is rancid, food is delicious, and so on. Before the rules can be evaluated, the inputs must be fuzzified according to each of these linguistic sets. For example, to what extent is the food really delicious? The Fig 3.14 below shows how well the food at our hypothetical restaurant (rated on a scale of 0 to 10) qualifies, (via its membership function), as the linguistic variable "delicious." In this case, we rated the food as an 8, which, given our graphical definition of delicious, corresponds to μ = 0.7 for the "delicious" membership function.

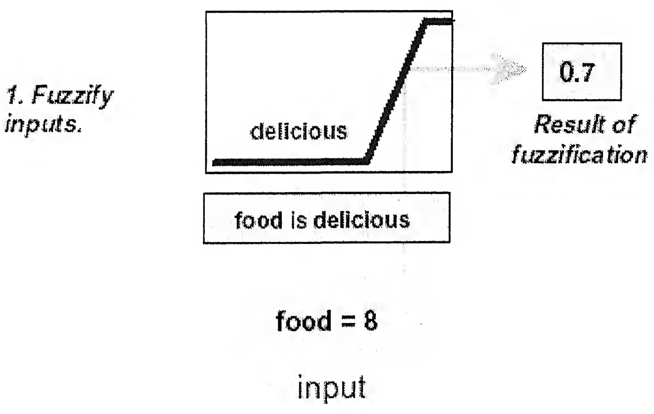


Fig 3.14 Membership Function For Variable Delicious

(The compliment to the chef would be "your food is delicious to the degree 0.7.")

In this manner, each input is fuzzified over all the qualifying membership functions required by the rules.

Step 2. Apply Fuzzy Operator

Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth-value.

As is described in the section on fuzzy logical operations, any number of well-defined methods can fill in for the AND operation or the OR operation. In the Fuzzy Logic Toolbox, two built-in AND methods are supported: *min* (minimum) and *prod* (product). Two built-in OR methods are also supported: *max* (maximum), and the probabilistic OR method *probor*. The probabilistic OR method (also known as the algebraic sum) is calculated according to the equation

$probor\ (a, b) = a + b - ab$

In addition to these built-in methods, you can create your own methods for AND and OR by writing any function and setting that to be your method of choice. There will be more information on how to do this later.

Fig 3.15 below is an example of the OR operator *max* at work. We're evaluating the antecedent of the rule 3 for the tipping calculation. The two different pieces of the antecedent (service is excellent and food is delicious) yielded the fuzzy membership values 0.0 and 0.7 respectively. The fuzzy OR operator simply selects the maximum of the two values, 0.7, and the fuzzy operation for rule 3 is complete. If we were using the probabilistic OR method, the result would still be 0.7 in this case.

Step 3. Apply Implication Method

Before applying the implication method, we must take care of the rule's weight. Every rule has a *weight* (a number between 0 and 1), which is applied to the number given by the antecedent. Generally this weight is 1 (as it is for this example) and so it has no effect at all on the implication process. From time to time you may want to weight one rule relative to the others by changing its weight value to something other than 1. Once proper weighting has been assigned to each rule, the implication method is implemented as in Fig 3.16.

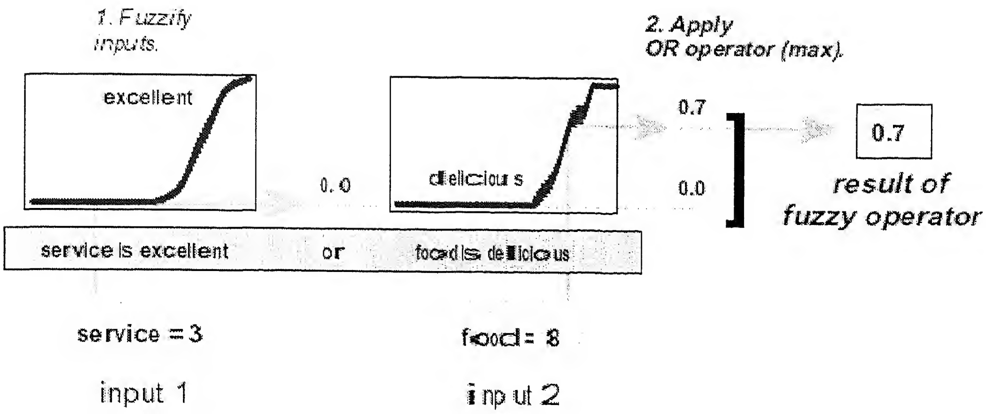


Fig 3.15 Probabilistic OR operator

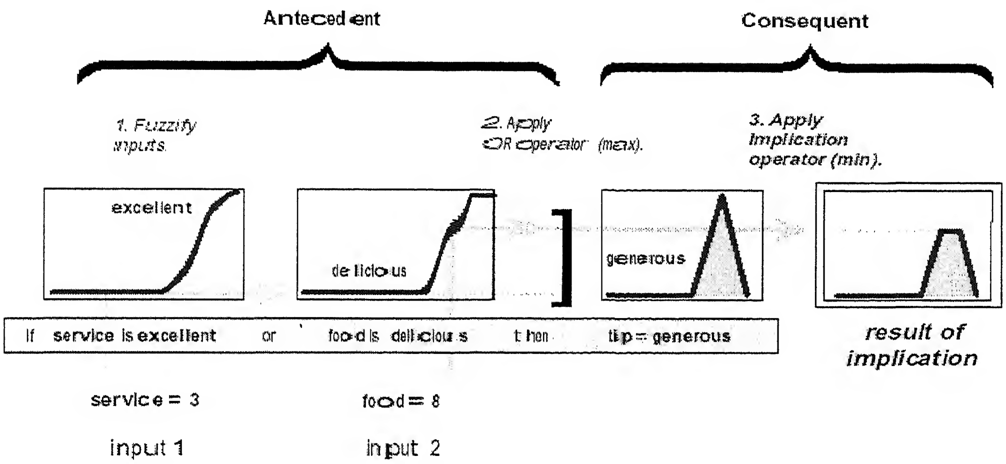


Fig 3.16 Implication Method

A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule. Two built-in methods are supported, and they are the same functions that are used by the AND method: *min* (minimum), which truncates the output fuzzy set, and *prod* (product), which scales the output fuzzy set.

Step 4. Aggregate All Outputs

Since decisions are based on the testing of all of the rules in an FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are

combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

Notice that as long as the aggregation method is commutative (which it always should be), then the order in which the rules are executed is unimportant. Three built-in methods are supported: *max* (maximum), *probor* (probabilistic or), and *sum* (simply the sum of each rule's output set).

In the Fig 3.17 below, all three rules have been placed together to show how the output of each rule is combined, or aggregated, into a single fuzzy set whose membership function assigns a weighting for every output (tip) value.

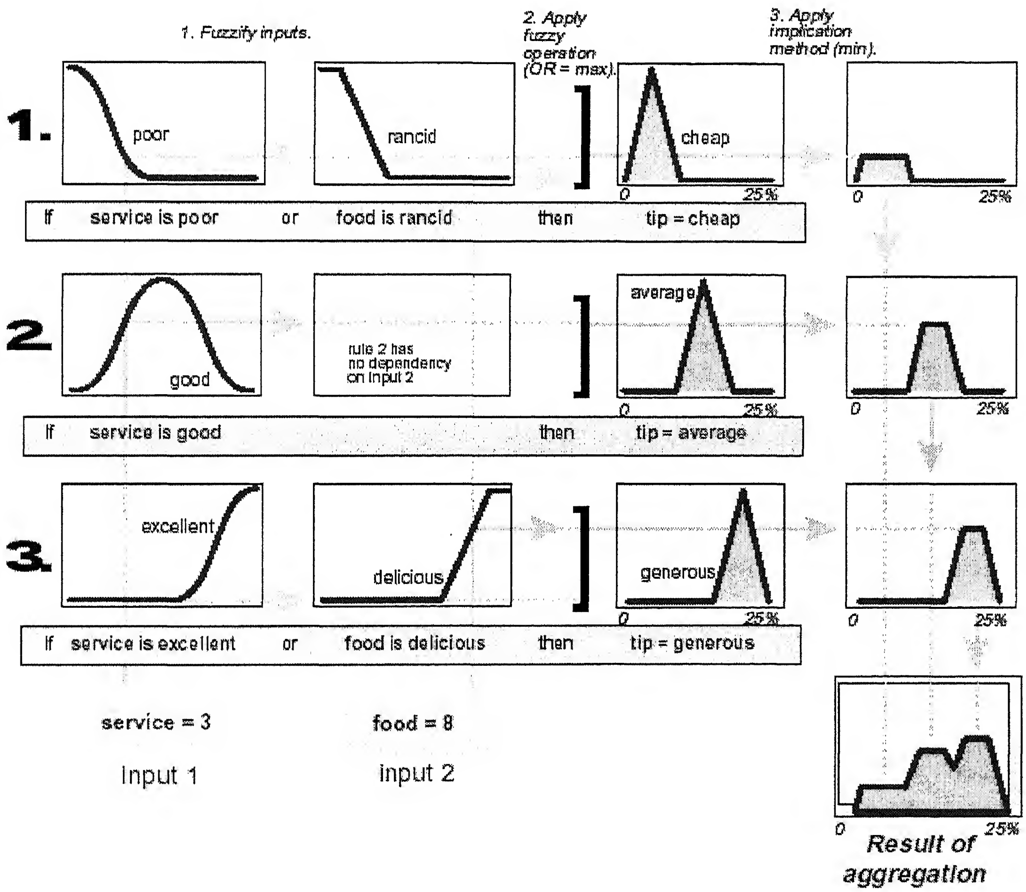


Fig 3.17 Aggregation Method

Step 5. Defuzzify

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve as shown in Fig 3.18 below. There are five built-in methods supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum.

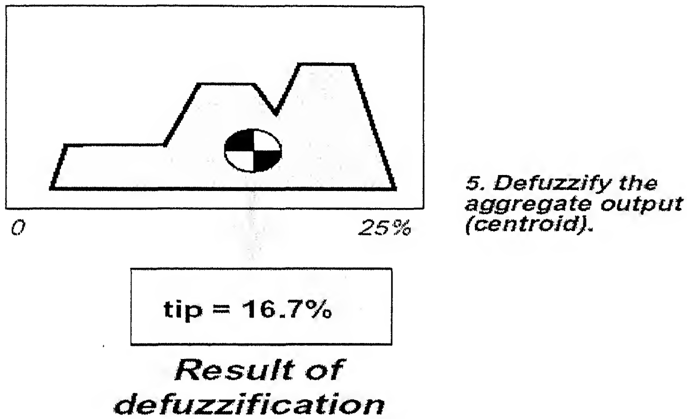


Fig 3.18 Centriod Method for Defuzzification

3.5.2 The Fuzzy Inference Diagram

The fuzzy inference diagram in Fig 3.19 below is the composite of all the smaller diagrams we have been looking at so far in this section. It simultaneously displays all parts of the fuzzy inference process we have examined. Information flows through the fuzzy inference diagram as shown below.

Notice how the flow proceeds up from the inputs in the lower left, then across each row, or rule, and then down the rule outputs to finish in the lower right. This is a very compact way of showing everything at once, from linguistic variable fuzzification all the way through defuzzification of the aggregate output.

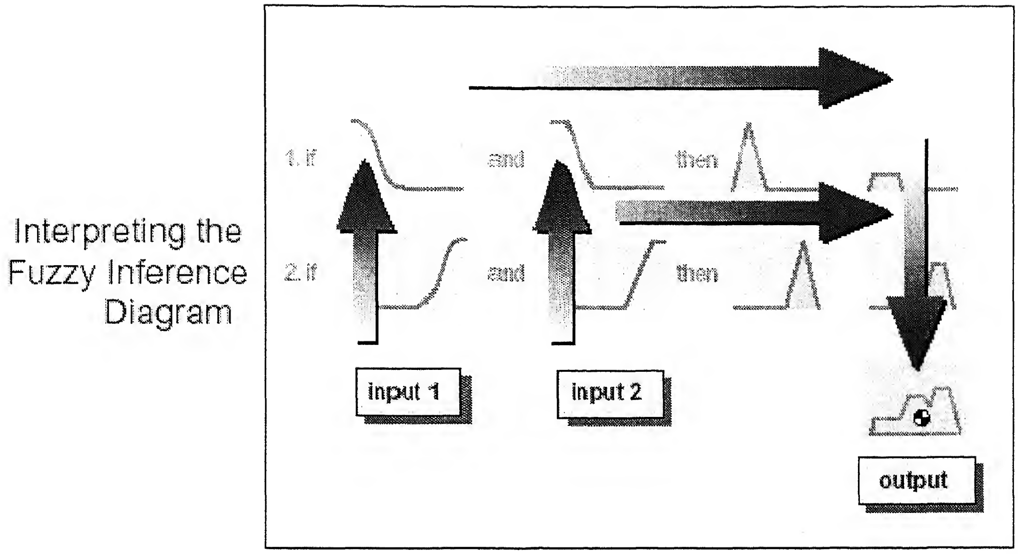


Fig 3.19 Fuzzy Inference Diagram

3.5.3 Sugeno-Type Fuzzy Inference

The fuzzy inference process we've been referring to so far is known as Mamdani's fuzzy inference method. It's the most commonly seen fuzzy methodology. In this section we discuss the so-called Sugeno, method of fuzzy inference. It is similar to the Mamdani method in many respects. In fact the first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani type of fuzzy inference and Sugeno-type is that the output membership functions are only linear or constant for Sugeno-type fuzzy inference.

A typical fuzzy rule in a *zero-order Sugeno fuzzy model* has the form

if x is A and y is B then $z = k$

Where A and B are fuzzy sets in the antecedent, while k is a crisply defined constant in the consequent. When the output of each rule is a constant like this, the similarity with Mamdani's method is striking. The only distinctions are the fact that all output membership functions are singleton spikes, and the implication and aggregation methods are fixed and cannot be edited. The implication method is simply multiplication, and the aggregation operator just includes all of the singletons.

The Fig 3.20 below shows the fuzzy tipping model developed in previous sections of this manual adapted for use as a zero-order Sugeno system. Fortunately it is frequently the case that singleton output functions are completely sufficient for a given problem's

needs.

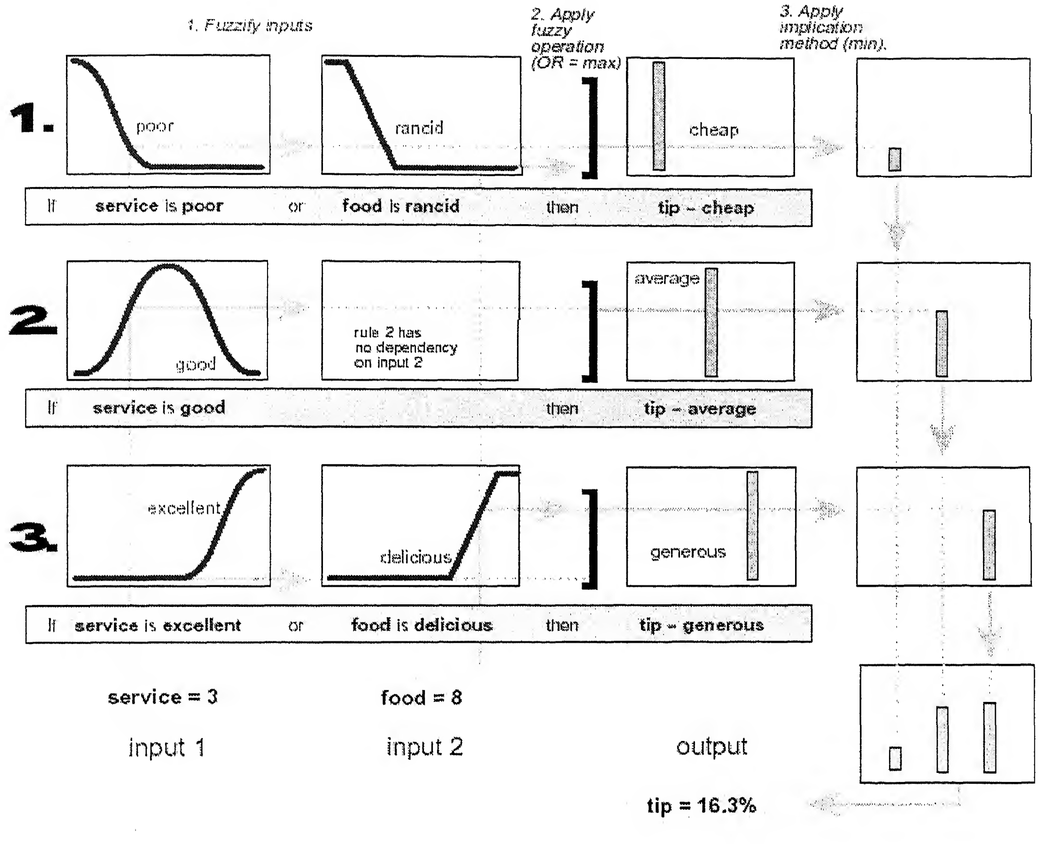


Fig 3.20 Sugeno fuzzy system for the tipping problem

The Mamdani implication method has been used in design of the Expert based system for RSP and the Sugeno method has been used in design of the data based system. Hence this has been discussed in detail in this chapter.

CHAPTER 4
DECISION TREES (ID3)

4.1 Overview of Decision Trees

Decision tree learning is a method for approximating discrete-valued target functions and is one of the most widely used and practical methods for inductive inference. Decision tree learning is generally best suited to problems with the following characteristics: Attribute-value pairs represent instances.

Instances are described by a fixed set of attributes (e.g., temperature) and their values (e.g., hot).The easiest situation for decision tree learning occurs when each attribute takes on a small number of disjoint possible values (e.g., hot, mild, cold). Extensions to the basic algorithm allow handling real-valued attributes as well (e.g., a floating point temperature). The target function has **discrete output values**. A decision tree assigns a classification to each example. Simplest case exists when there are only two possible classes (**Boolean classification**). Decision tree methods can also be easily extended to learning functions with more than two possible output values. A more substantial extension allows learning target functions with real-valued outputs, although the application of decision trees in this setting is less common. Disjunctive descriptions may be required. Decision trees naturally represent disjunctive expressions.

4.2 Decision Tree Representation

A **decision tree** is an arrangement of tests that prescribes an appropriate test at every step in an analysis. In general, decision trees represent a disjunction of conjunctions of constraints on the attribute-values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests and the tree itself to a disjunction of these conjunctions. More specifically, decision trees classify **instances** by sorting them down the tree from the **root node** to some **leaf node**, which provides the classification of the instance. Each node in the tree specifies a **test** of some **attribute** of the instance, and each **branch** descending from that node corresponds to one of the possible **values** for this attribute.

An instance is classified by starting at the root node of the decision tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated at the node on this branch and so on until a leaf node is reached.

4.3 Decision Tree Diagram

- Each nonleaf node is connected to a test that splits its set of possible answers into subsets corresponding to different test results.
- Each branch carries a particular test result's subset to another node.
- Each node is connected to a set of possible answers.

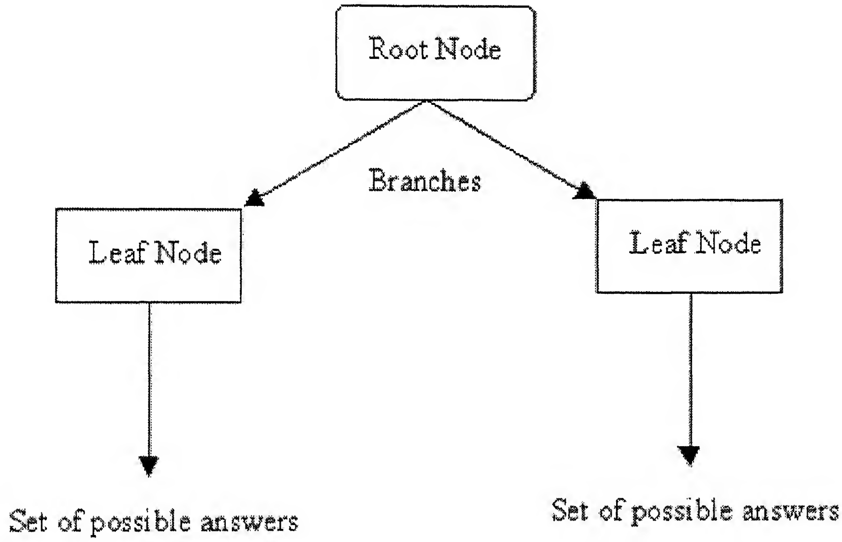


Fig 4.1 Tree Structure

4.4 Quinlan's ID3 Algorithm for Constructing a Decision Tree

General Form

Until each leaf node is populated by as homogeneous a sample set as possible. Select a leaf node with an inhomogeneous sample set. Replace that leaf node by a test node that divides the inhomogeneous sample set into minimally inhomogeneous subsets, according to an entropy calculation.

Specific Form

Examine the attributes to add at the next level of the tree using an entropy calculation. Choose the attribute that minimizes the entropy.

4.4.1Choosing an Attribute using the Entropy Formula

The central focus of the ID3 algorithm is selecting which attribute to test at each node in the tree.

Procedure:

See how the attribute distributes the instances.

Minimize the average entropy.

Calculate the average entropy of each test attribute and choose the one with the lowest degree of entropy .

4.4.2 Entropy Formulae

Entropy, a measure from information theory, characterizes the (im) purity, or homogeneity, of an arbitrary collection of examples.

Given:

n_b = the number of positive instances in branch b.

n_{bc} = the total number of instances in branch b of class c.

n_t = the total number of instances in all branches.

Probability

$$P_b = \text{Probability an instance on a branch } b \text{ is positive}$$

$$= \frac{\text{number of positive instances on branch}}{\text{total number of instances on branch}} = \frac{n_{bc}}{n_b}$$

If all the instances on the branch are positive, then $P_b = 1$ (homogeneous positive) If all the instances on the branch are negative, then $P_b = 0$ (homogeneous negative)

Entropy

$$\text{Entropy} = \sum_c -\left(\frac{n_{bc}}{n_b}\right)\log_2\left(\frac{n_{bc}}{n_b}\right)$$

$H = \sum - (P_b \log_2 (P_b))$ ----------------------------------4.1

Average Entropy (H)

$$\text{Or } H(C \mid A_k) = \sum_{j=1}^{M_k} P(a_k, j) * \left[-\sum_{i=1}^{N} P(C_i \mid a_{k,j}) * \log_2 P(C_i \mid a_{k,j}) \right] \text{--------4.2}$$

Where,

$H(C|A_k)$ = entropy of classification property of attribute A_k.

$P(a_{k,j})$ = probability of attribute k being at value j.

$P(C_i \mid a_{k,j})$ = probability that the class value is C_i when attribute k is at its j^{th} value.

M_k = Total number of values for attribute A_k : j= 1, 2, …, M_k.

N= total number of different classes; i =1, 2… N.

K= total number of attributes; k =1, 2… K.

4.5 Use of ID3 in developing rule base for RSP

Two different sets of rule bases were formed for implementation of the system.

- **Rule Base from Expert Advice**: The rule base was formulated on the knowledge acquired from engineers and worker at the steel plant right from the supervisors downward during visit at the factory premises.
- **Data based:** This rule base was formed with the data made available from plant. The rules were formulated by assigning discrete values of high, medium and low as per the numeric values taken by them.

4.5.1Rule Base from Expert Advice

As the available data at the start of project was very few in number and insufficient to develop the rule base, the development of expert system was started with rule base made on the advice, and experience of the experts. As the number of input parameters was high (10) and each of the attributes above took values like very high, high, medium, low and very low the number of rules to get the green bulk density would be enormous ($3^8 * 5^2$ to be precise as PMT and GBD had 5 values and remaining 8 attributes had 3 values). Thus the rule sets

were formulated so as to reduce the memory intensive computing that would be involved otherwise and the entire process of brick manufacture was broken up into sub processes as shown in the table 4.1 below,

Sub process1	Inputs-3	Output-1
	CTI	GT
	GHDT	
	RT	
Sub process2	Inputs-3	Output-1
	GT	MIX TEMP
	MIXER TEMP	
	MIXING TIME	
Sub process3	Inputs-3	Output-1
	MIX TEMP	PMT
	ST	
	No of ROLLS	
Sub process4	Inputs-4	Output-1
	PMT	GBD
	PITCH	
	GRAPHITE	
	FP	
Sub process5	Inputs-3	Output-1
	GBD	CCS
	PITCH	
	TEMPERING TEMP	
Sub process6	Inputs-2	Output-1
	GBD	CP
	PITCH	
Sub process7	Inputs-3	Output-1
	CP	SCR
	Carbon Content	
	Chemical Purity	
Sub process8	Inputs-3	Output-1
	GBD	OXI RESISTANCE
	Carbon Content	
	Metal Powder	
Sub process9	Inputs-3	Output-1
	GBD	HMOR
	Carbon Content	
	CCS	
	Metal Powder	

Table 4.1 Sub processes for rule generation

4.5.2 Sub Process-1

Let us go on to solve the expert base data set on GT using ID3 algorithm and construct the tree structure and generate the rules.

Entropy calculations

The data set based on expert advice is shown in table 4.2 of AppendixA in which we have input attributes as CTI, GHDT and RT, hence k=3,

There are 5 classes (very_high, high, medium, low & very_low) for the output, hence N=5.

There are 3 classes each for the inputs (high, medium & low), M_1, M_2 and M_3 =3

Using equation 4.2 we can compute the entropy for each attribute, starting with CTI we proceed as follows,

P (CTI = high) = 9/27 =1/3

P (CTI = medium) = 9/27 =1/3

P (CTI = low) = 9/27 =1/3

P (GT = very high | CTI= high) = 1/9

P (GT=high | CTI=high) = 2/9

P (GT=medium | CTI=high) = 4/9

P (GT=low | CTI=high) = 1/9

P (GT=very low | CTI=high) = 1/9

P (GT=very high | CTI=medium) = 1/9

P (GT=high | CTI=medium) = 2/9

P (GT=medium | CTI=medium) = 3/9

P (GT=low| CTI=medium) = 2/9

P (GT=very low | CTI=medium) = 1/9

P (GT=very high | CTI=low) = 0/9

P (GT= high | CTI=low) = 1/9

P (GT=medium | CTI=low) = 3/9

P (GT=low | CTI=low) = 3/9

P (GT=very low | CTI=low) = 2/9

Similarly corresponding values for GHDT and RT are computed and we have by substituting in the equation 4.2,

H (GT | CTI) = 0.6168139456208

H (GT | GHDT) = 0.4873750559284

H (GT | RT) = 0.5330855183249

Thus we find that GHDT has the minimum average entropy and hence a decision tree with root node as GHDT is constructed as shown in Fig4.2

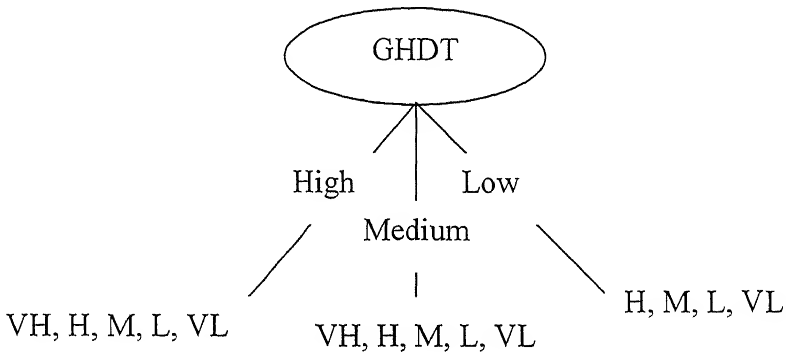


Fig4.2 Incomplete Branching

Listed below each attribute value in the fig4.2 are the classes for Grain temp associated with that particular branching of decision tree, for example under GHDT equals High we have all classes for GT where as under GHDT equals Low we have all classes except very high. Since all three branches result in an inconclusive classification, we have to branch in yet another attribute.

Now we must partition table 4.2 according to attribute GHDT. This results in three subtables (one each for GHDT equals high, medium and low). The subtable for GHDT equals high is shown in Table (4.2A). Once again we compute the entropy for each attribute in the three subtables. The other two sub tables are shown in Table (4.2 B & C).

CTI	RT	GRAIN TEMP
HIGH	HIGH	VERY HIGH
HIGH	MEDIUM	HIGH
HIGH	LOW	MEDIUM
MEDIUM	HIGH	VERY HIGH
MEDIUM	MEDIUM	HIGH
MEDIUM	LOW	MEDIUM
LOW	HIGH	HIGH
LOW	MEDIUM	MEDIUM
LOW	LOW	LOW

Table (4.2A) Classification when GHDT equals High

CTI	RT	GRAIN TEMP
HIGH	HIGH	HIGH
HIGH	MEDIUM	MEDIUM
HIGH	LOW	MEDIUM
MEDIUM	HIGH	HIGH
MEDIUM	MEDIUM	MEDIUM
MEDIUM	LOW	LOW
LOW	HIGH	MEDIUM
LOW	MEDIUM	MEDIUM
LOW	LOW	LOW

Table (4.2B) Classification when GHDT equals Medium

CTI	RT	GRAIN TEMP
HIGH	HIGH	MEDIUM
HIGH	MEDIUM	LOW
HIGH	LOW	VERY LOW
MEDIUM	HIGH	MEDIUM
MEDIUM	MEDIUM	LOW
MEDIUM	LOW	VERY LOW
LOW	HIGH	LOW
LOW	MEDIUM	VERY LOW
LOW	LOW	VERY LOW

Table (4.2B) Classification when GHDT equals Low

Entropy Calculations GHDT (High)

Using equation 4.2 we get,

H(GT| CTI)=3/9{ 3*[1/3*log(1/3)]+[1/3*log(1/3)+2/3log(2/3)]

+ 3*[1/3*log(1/3)]}= 0.477121254719662437295

H(GT| RT)=3/9{ 3*[1/3*log(1/3)]+ 3*[1/3*log(1/3)]

+[1/3*log(1/3)+2/3log(2/3)]}= 0.2764345909436749738

Here we see the RT has the lowest entropy. Hence it will form the next node and since only one attribute is left that of CTI which will form the terminal node and the tree structure will be complete as in fig 4.3. and fig 4.4.

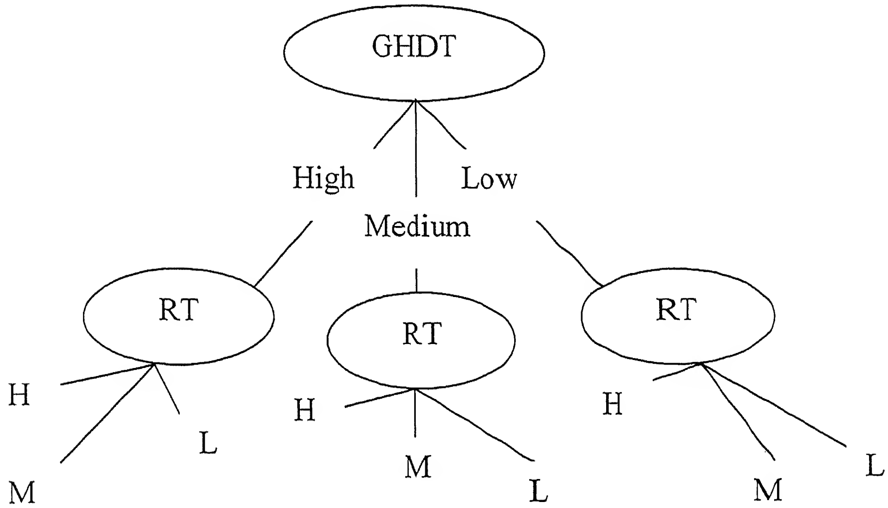


Fig4.3 Partially complete Branching

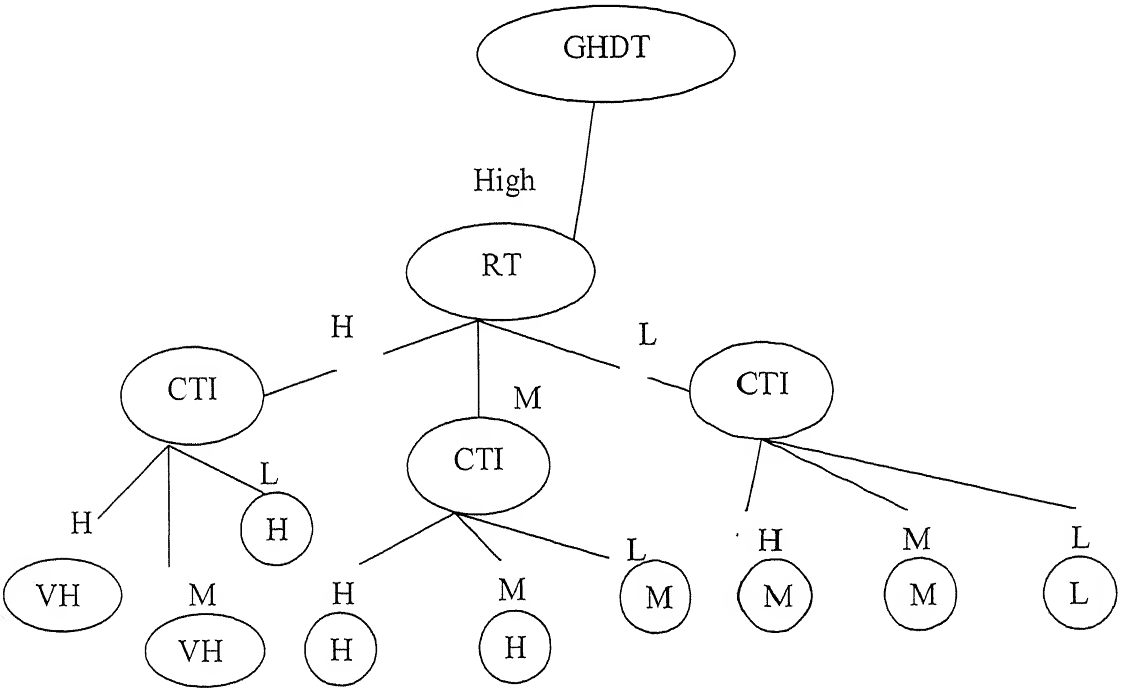


Fig4.4 Complete Branching for GHDT (High)

Entropy Calculations GHDT (Medium)

Using equation 4.2 we get,

H(GT| CTI)=3/9{ 3*[1/3*log(1/3)]+[1/3*log(1/3)+2/3log(2/3)]

+ 3*[1/3*log(1/3)]}= 0.34333014553567079497

H(GT| RT)=3/9{ 3*[1/3*log(1/3)]+ 3*[1/3*log(1/3)]

+[1/3*log(1/3)+2/3log(2/3)]}= 0.18428972729578331587

Entropy Calculations GHDT (Medium)

Using equation 4.2 we get,

H(GT| CTI)=3/9{ 3*[1/3*log(1/3)]+[1/3*log(1/3)+2/3log(2/3)]

+ 3*[1/3*log(1/3)]}= 0.41022570012766661613

H(GT| RT)=3/9{ 3*[1/3*log(1/3)]+ 3*[1/3*log(1/3)]

+[1/3*log(1/3)+2/3log(2/3)]}= 0.18428972729578331587

Thus we see that for GHDT Medium and Low also we get RT having lower entropy than CTI. Software has been used to calculate the average entropy using Quinlan's ID3 Algorithm and to generate the decision trees for the data for the processes attached as Table 4.3 to 4.10 in AppendixA. The trees generated for data sets from Sub processes 1 to 9 has been attached as fig 4.5 to 4.13 in Appendix B.

4.5.3 Generation of Rule Base from Available Data

The Data available at the start from plant was only 45 complete sets up to the Green BD stage. Subsequently further data was provided up to 86 sets. The Data Based Rules have been formed only up to the green BD stage as at present no data is available for Sub processes beyond GBD stage. The rules were made with the following inputs and Green BD as the output:

- Combustion Chamber Temperature
- Grain Heater Drum Temperature
- Retention Time in Grain Heater
- Mixer temperature
- Mixing time
- Mix temperature
- Storage time
- Number of rolls
- Press Mix Temperature
- Forming pressure

The other parameters like the granulometry, pitch quantity and graphite quantity in the data was fixed and remained constant. Thus the same were not used as input. The 86 data sets shown in table 4.11A of Appendix A was converted to a table of very high, high, medium, low and very low as the case may be by taking calculating the mean and standard deviation(SD) of the attribute values, after which values below 2 SD were assigned to low, within 2 SD to Medium and all above 2 SD to High, and the same is shown in table 4.11B of Appendix A. The ID3 algorithm was applied to this data set in table 4.11B. The tree structure obtained for the data is shown in fig 4.14of Appendix B.

The results obtained from the data was in variation with the expert opinion because the data collected had very high noise content and secondly the quantity of data sets was far too small to capture major portion of the feature space. It was later admitted by the plant staff that the data could be noisy due lack of proper instrumentation and ignorance on part of the ground workers on the importance of collecting accurate data. To observe a pattern with in the data at least 3000 to 4000 sets of data will be required. The system can be made robust and reliable only once it is able to capture 90 to 95 % of the feature space. For the 86 data sets the ID3 algorithm generated 66 sets of rules that can be read out from the tree structure attached as per fig 4.14 of Appendix B.

CHAPTER 5
CLASSIFICATION AND REGRESSION TREES

5.1 Overview of cart

CART is a robust data-mining and data-analysis tool that automatically searches for important patterns and relationships and quickly uncovers hidden structure even in highly complex data. A decision tree is a flow chart or diagram representing a classification system or predictive model. The tree is structured as a sequence of simple questions, and the answers to these questions trace a path down the tree. The end point reached determines the classification or prediction made by the model.

The CART methodology is technically known as binary recursive partitioning. The process is binary because parent nodes are always split into exactly two child nodes and recursive because the process can be repeated by treating each child node as a parent. The key elements of a CART analysis are a set of rules for:

- Splitting each node in a tree,
- Deciding when a tree is complete, and
- Assigning each terminal node to a class outcome (or predicted value for regression).

5.2 Splitting Rules

To split a node into two child nodes, CART always asks questions that have a "yes" or "no" answer. For example, in the tree developed for RSP using the 117 data sample shown in fig 5.1 below,

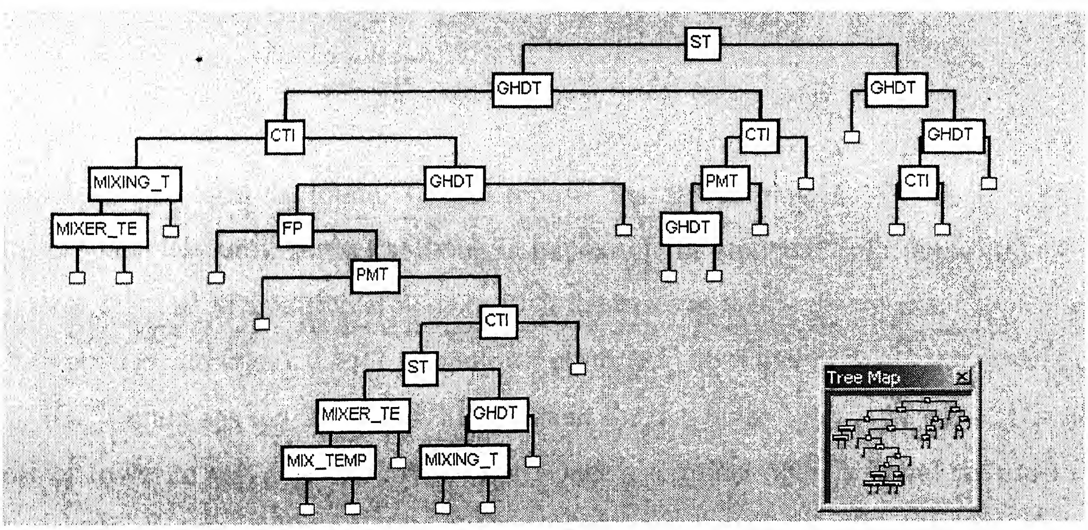


Fig 5.1 Tree structure developed for RSP using 117 data sets.

Node 1 was split on ST, a case goes left if ST <= 58.500, i.e. the root, or parent node is split into two branches with "yes" cases going to the left child node and "no" cases to the right, and Node 2 was split on GHDT, a case goes left if GHDT <= 537.500, similarly Node 3 was split on CTI, a case goes left if CTI <= 342.000.

5.3 Choosing a Split

Next step is to rank order each splitting rule on the basis of a quality-of split criterion. The default criterion used in CART is the GINI rule; essentially a measure of how well the splitting rule separates the classes contained in the parent node. Five alternative criteria are also available for classification trees and two criteria for regression trees. In addition, to deal more effectively with select data patterns, CART also offers splits on linear combination of continuous predictor variables. For example shown in Fig 5.2 below is partial portion of the tree developed up to level 3 in the case of RSP data the best quality slip obtained at Node 1 slips all 117 data, Node 2 slips 82 data sets , Node 3 splits 62 sets is done using GINI rule.

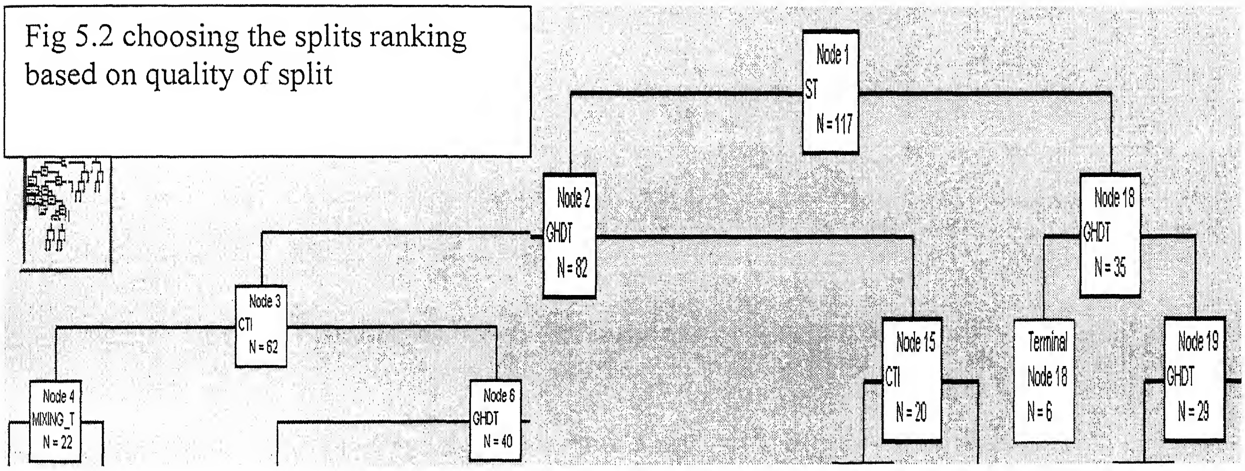


Fig 5.2 choosing the splits ranking based on quality of split

5.4 Class Assignment

Once a best split is found, CART repeats the search process for each child node, continuing recursively until further splitting is impossible or stopped. Splitting is impossible if only one case remains in a particular node or if all the cases in that node are exact copies of each other (on predictor variables). CART also allows splitting to be stopped for several other reasons, including that a node has too few cases. (The default for this lower limit is 10 cases, but may be set higher or lower to suit a particular analysis). Once a terminal node is found we must decide how to classify all cases falling within it. One simple criterion is the plurality rule: the group

with the greatest representation determines the class assignment. For example shown in Fig 5.3 below is the case for class 1 or very low GBD in red square boxes, the degree of redness gives the class assignment for it. The best to worst case changing gradually from red to blue, in the fig Below only three nodes have cases of very low GBD.

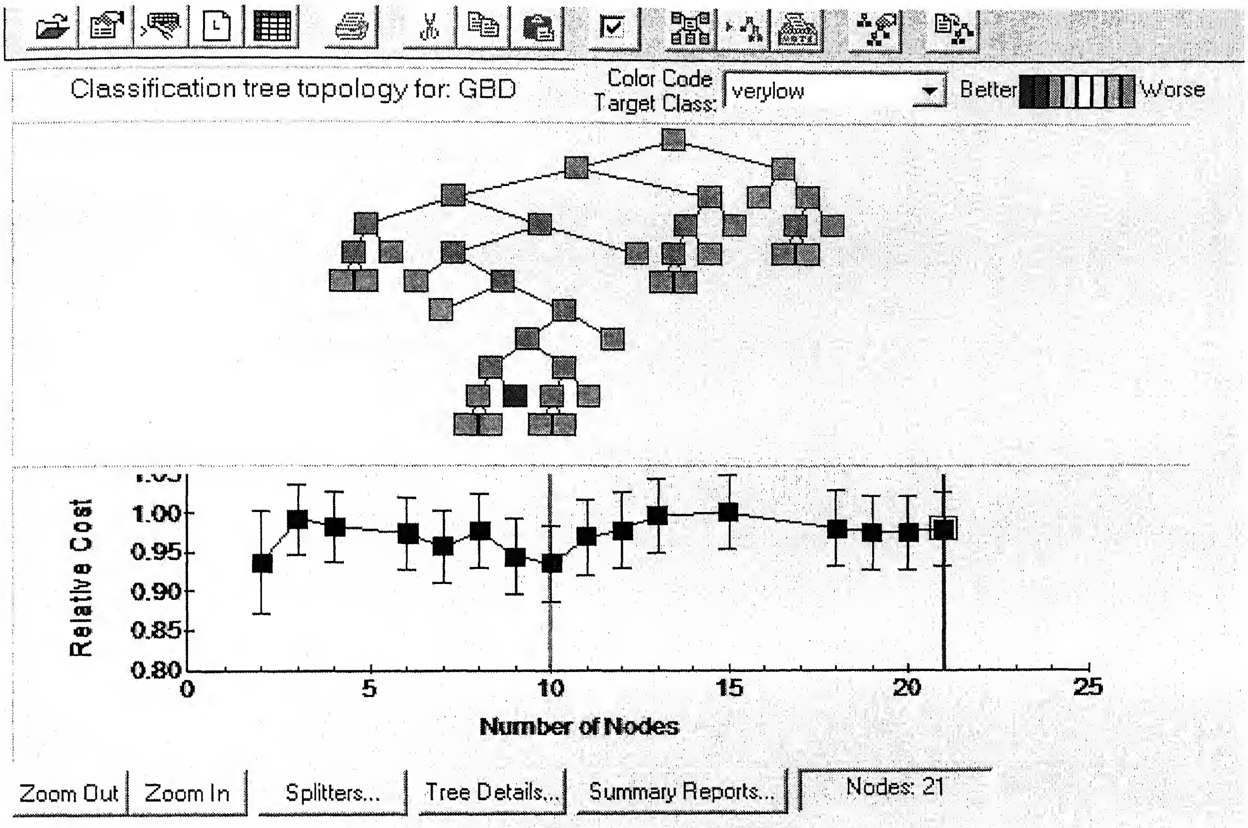


Fig 5.3 Class Assignment based on Number of cases in a Node

5.5 Testing

Once the maximal tree is grown and a set of sub-trees is derived from it, CART determines the best tree by testing for error rates or costs. With sufficient data, the simplest method is to divide the sample into learning and test sub-samples. The learning sample is used to grow an overly large tree. The test sample is then used to estimate the rate at which cases are misclassified (possibly adjusted by misclassification costs). The misclassification error rate is calculated for the largest tree and also for every sub-tree. The best sub-tree is the one with the lowest or near-lowest cost, which may be a relatively small tree. Some studies, as the one in our case of RSP data, where there are insufficient data samples to allow a good-sized separate test sample, CART employs the computer-intensive technique of cross validation.

5.6 Cross Validation

In such cases, CART grows a maximal tree on the entire learning sample. This is the tree that will be pruned back. CART then proceeds by dividing the learning sample into 10 roughly-equal parts, each containing a similar distribution for the dependent variable, in the case of 117 data sets each sample would have 12 samples with the last having 8 samples. CART takes the first 9 parts of the data, constructs the largest possible tree, and uses the remaining 1/10 of the data to obtain initial estimates of the error rate of selected sub-trees. The same process is then repeated (growing the largest possible tree) on another 9/10 of the data while using a different 1/10 part as the test sample. The process continues until each part of the data has been held in reserve one time as a test sample. The results of the 10 mini-test samples are then combined to form error rates for trees of each possible size; these error rates are applied to the tree based on the entire learning sample.

The upshot of this complex process is a set of fairly reliable estimates of the independent predictive accuracy of the tree. This means that we can know how well any tree will perform on completely fresh data—even if we do not have an independent test sample. Because the conventional methods of assessing tree accuracy can be wildly optimistic, cross validation is the method CART normally uses to obtain objective measures for smaller data sets.

5.7 Surrogate Splitter

CART handles missing values in the database by substituting "surrogate splitters," which are back-up rules that closely mimic the action of primary splitting rules. Suppose that, in a given model, CART splits data according to household income. If a value for income is not available, CART might substitute education level as a good surrogate.

The surrogate splitter contains information that is typically similar to what would be found in the primary splitter. Other products' approaches treat all records with missing values as if the records all had the same unknown value; with that approach all such "missing" are assigned to the same bin. In CART, each record is processed using data specific to that record; this allows records with different data patterns to be handled differently, which results in a better characterization of the data.

5.8 Special Features of CART

Easy Data Access

With a direct link to DBMS, CART provides easy access to over 80 different file formats and supports ODBC for both reading and writing. For example, you can import and export statistical analysis packages (e.g., SAS, SPSS), databases (e.g., Oracle, Informix), and spreadsheets (e.g., Excel, Lotus).

Automatic Rule Generation

With a click of a button one can, view if-then rules for any individual node or for the entire tree. To facilitate external classifying and scoring of new data, the splitting rules are formatted as C-compatible programming statements that can be exported as either a text file or cut and pasted into another application.

5.9 Generation of Cart Tree for RSP data.

The opening screen of the cart programme displays as under, which is very similar to the windows icons, the function of each of the buttons is explained below

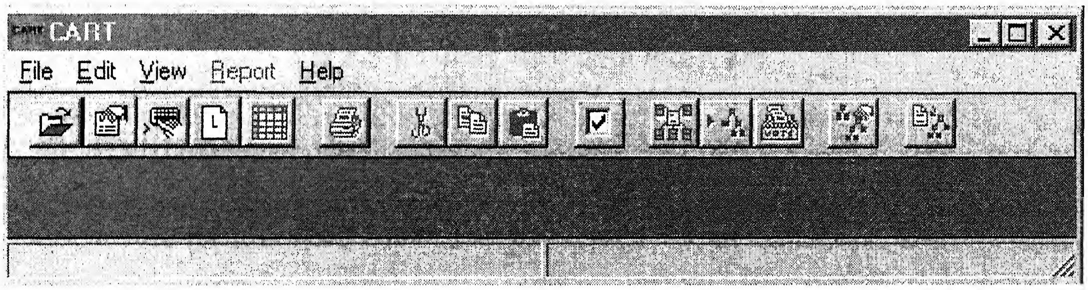


Fig 5.4 the opening screen of Cart.

The most commonly-used commands have corresponding toolbar icons. The following icons are most frequently used:

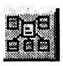
 Opening an existing file for analysis

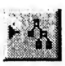
 Submitting a text file of CART commands for batch processing

 Turning command-line entry mode on or off

 Opening a log of processed CART commands

 Setting reporting, random number and directory options

 Specifying model variables and parameters that control analysis

 Starting CART analysis of current model

When the open button is pressed one is prompted to load the data file, in which the RSP data set in Excel format is loaded to give the following window in Fig 5.5 below,

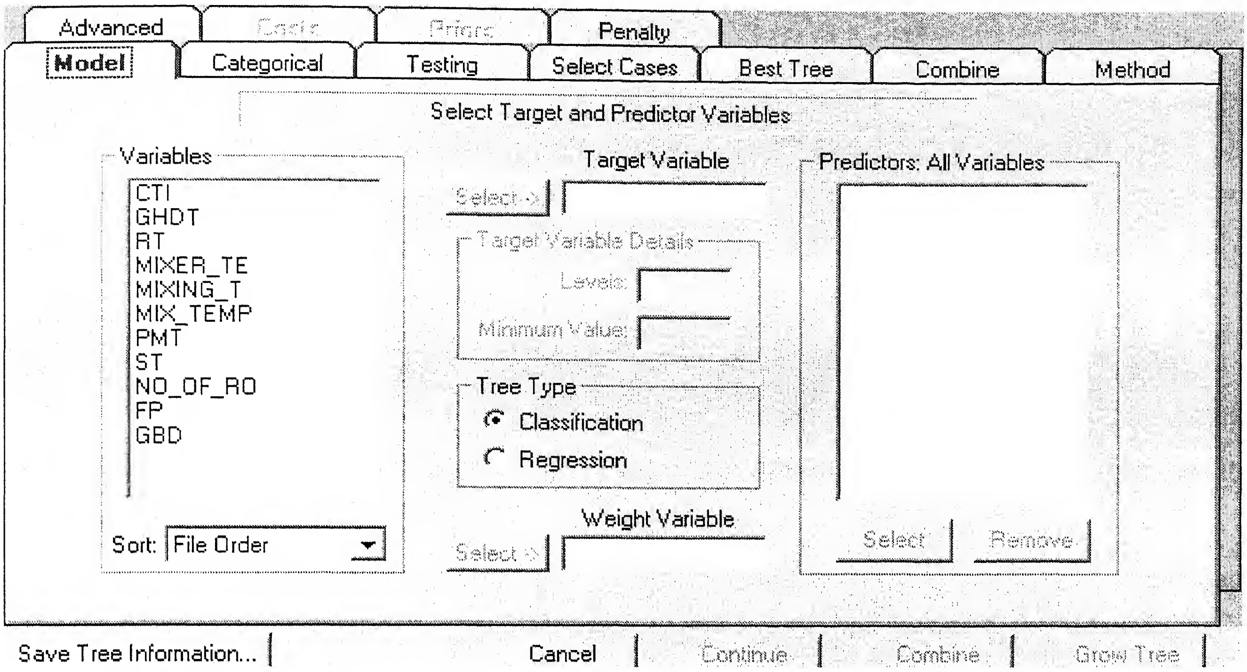


Fig 5.5 Loading RSP data into Cart

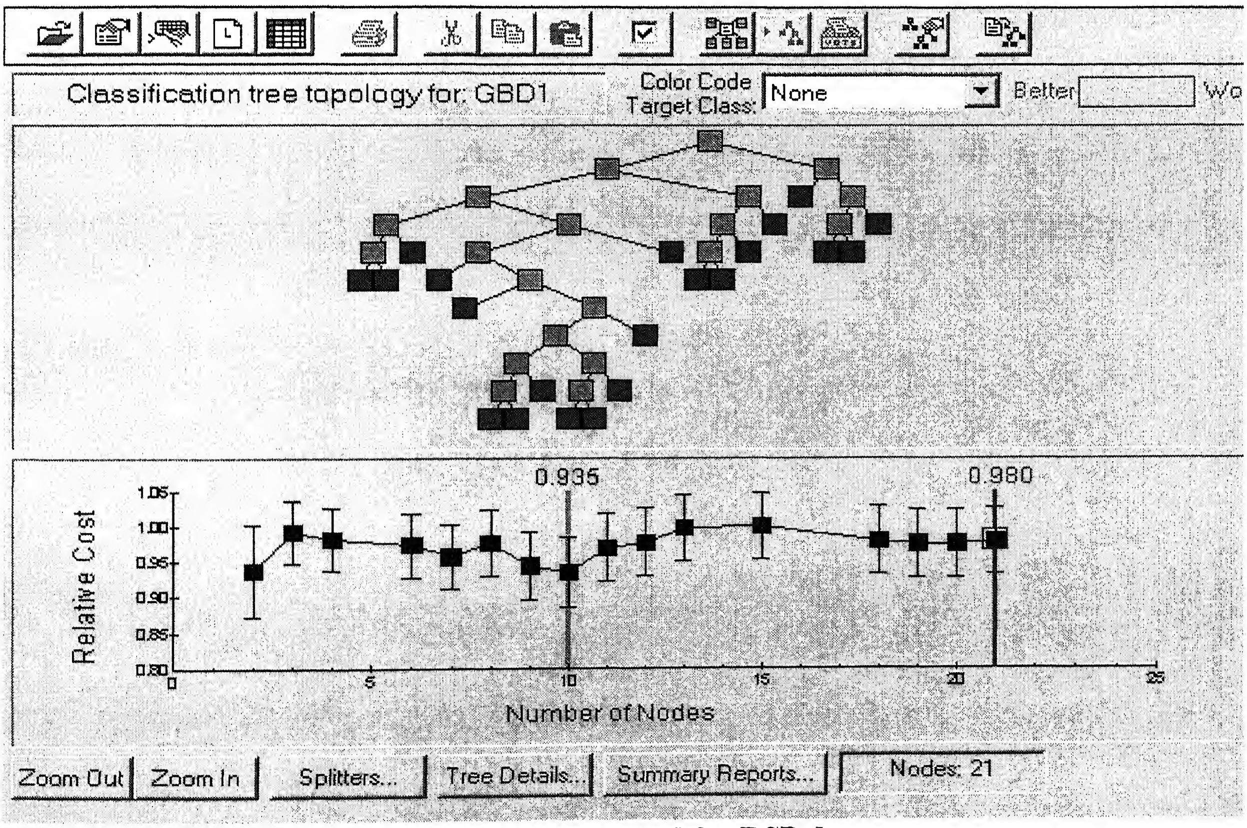


Fig 5.6 Tree generated for RSP data

The target variable selected is GBD and the rest are the predictors. The number of levels is set to 5 for the target variable corresponding to 1 to 5 being very low, low, medium, high and very high, hence minimum value is set to 1. Then the grow tree button is pressed to give the following screen as in Fig 5.6 above.

5.10 Results obtained and use of Cart analysis in RSP data

Since the data collected from the plant was noisy in nature and not in adequate quantity hence the results obtained did not completely match with the expert opinion, which was adequately acknowledged by the RSP staff also. The salient features are discussed below,

The rule generated by cart is attached as per Appendix D. The rules generated are only 21 in number for the given data as compared to 66 rules generated by ID3, and 243 rules generated by matlab, for the same amount of data. However the ranges for high, medium, low are in general agreement with that given to us by the experts with minor changes.

The variable importance as deduced by Cart was not in agreement with the expert opinion due to reasons just brought earlier. The comparison of variable importance and the other out puts obtained by Cart Analysis is attached as Fig 5.7 & 5.8 of Appendix E.

CHAPTER 6

BAYESIAN NETWORKS AND PROBABILTY PREDICTIONS

6.1 Overview of Bayesian networks

The essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence. In other words, it allows us to combine new data with our existing knowledge or expertise. Mathematically, Bayes' rule states

$$Posterior\_\Pr obabilty = \frac{Conditional\_\Pr obabilty * Aprior\_\Pr obablity}{\Pr obability}$$

Or, in symbols, $P(R = r \mid e) = \dfrac{P(e \mid R = r)P(R = r)}{P(e)}$

Where P(R=r|e) denotes the probability that random variable R has value r given evidence e.

The denominator is just a normalizing constant that ensures the posterior adds up to 1; it can be computed by summing up the numerator over all possible values of R, i.e.,

$$P\ (e) = P(R=0, e) + P(R=1, e) + \ldots = \text{sum } P(e \mid R=r)\ P(R=r)$$

For complicated probabilistic models, computing the normalizing constant P (e) is computationally intractable (because there are an exponential or even infinite number of values of "R" to sum over); graphical models help simplify these .putations so that they become feasible.

Graphical Models

Graphical models are a marriage between probability theory and graph theory. y provide a natural tool for dealing with two problems that occur throughout lied mathematics and engineering -- uncertainty and complexity -- and in

particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity -- a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.

6.2.1 Types of Graphical Models

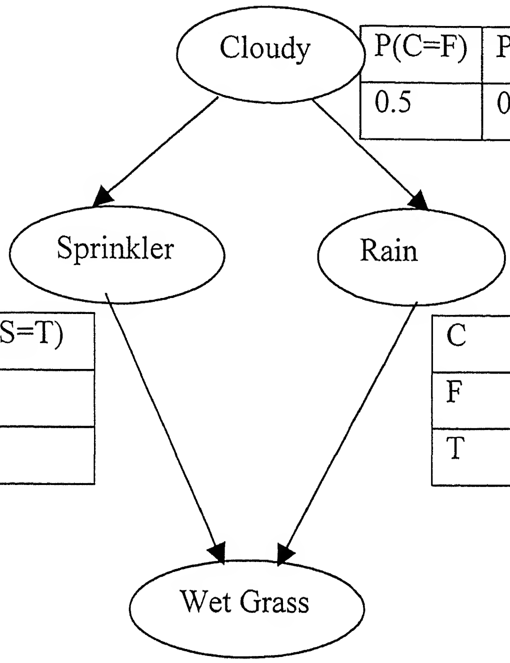
Graphical models are graphs in which nodes represent random variables, and the lack of arcs represents conditional independence assumptions. **Undirected graphical models, also called Markov Random Fields (MRFs)**, have a simple definition of independence: two (sets of) nodes A and B are conditionally independent given a third set, C, if all paths between the nodes in A and B are separated by a node in C. By contrast, **directed graphical models, also called Bayesian Networks or Belief Networks (BNs)**, have a more complicated notion of independence, which takes into account the directionality of the arcs. (Note that, despite the name, Bayesian networks do not necessarily imply a commitment to Bayesian methods; rather, they are so called because they use Bayes' rule for inference). Undirected graphical models are more popular with the physics and vision communities, and directed models are more popular with the AI and statistics communities.

6.3 Directed Graphical Models

6.3.1 Bayesian Networks

Although directed models have a more complicated notion of independence than undirected models, they do have several advantages. The most important is that one can regard an arc from A to B as indicating that A

``causes'' B. This can be used as a guide to construct the graph structure. In addition, directed models can encode deterministic relationships, and are easier to learn (fit to data).



P(C=F)	P(C=T)
0.5	0.5

S	P (S=F)	P (S=T)
F	0.5	0.5
T	0.9	0.1

C	P(C=F)	P(C=T)
F	0.8	0.2
T	0.2	0.8

S	R	P (W=F)	P (W=T)
F	F	1	0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Fig 6.1 Graph Structure and Conditional Probability Table (CPT)

In addition to the graph structure, it is necessary to specify the parameters of the model. For a directed model, we must specify the Conditional Probability Distribution (CPD) at each node. If the variables are discrete, this can be represented as a Conditional Probability table (CPT),

which lists the probability that the child node takes on each of its different values for each combination of values of its parents. Consider the following example in the Fig 6.1 above, in which all nodes are binary, i.e., have two possible values, which we will denote by T (true) and F (false). We see that the event "grass is wet" (W=true) has two possible causes: either the water sprinker is on (S=true) or it is raining (R=true). For example, we see that Pr (W=true | S=true, R=false) = 0.9 (second row), and hence, Pr (W=false | S=true, R=false) = 1 - 0.9 = 0.1, since each row must sum to one. Since the C node has no parents, its CPT specifies the prior probability that it is cloudy (in this case, 0.5).

6.3.1.1 Conditional Independence

The simplest conditional independence relationship encoded in a Bayesian network can be stated as follows: a node is independent of its ancestors given its parents, where the ancestor/parent relationship is with respect to some fixed topological ordering of the nodes. By the chain rule of probability, the joint probability of all the nodes in the graph above is P(C, S, R, W) = P (C) * P (S|C) * P (R|C, S) * P (W|C, S, R)

by using conditional independence relationships, we can rewrite this as P(C, S, R, W) = P(C) * P (S|C) * P (R|C) * P (W|S, R). Where we were allowed to simplify the third term because R is independent of S given its parent C, and the last term because W is independent of C given its parents S and R. We can see that the conditional independence relationships allow us to represent the joint more compactly. Here the savings are minimal, but in general, if we had n binary nodes, the full joint would require 2^n space to represent, but the factored form would require n* 2^k) space to represent, where k is the maximum fan-in of a node. And hence fewer parameters make learning easier.

6.3.1.2 Inference

The most common task we wish to solve using Bayesian networks is probabilistic inference. For example, consider the water sprinkler network, and suppose we observe the fact that the grass is wet. There are two possible

likely? We can use Bayes' rule to compute the posterior probability of each explanation (where 0==false and 1==true).

$$P(S=1 \mid W=1) = \frac{P(S=1, W=1)}{P(W=1)} =$$
$$\frac{\sum_{c,r} P(C=c, S=1, R=r, W=1)}{P(W=1)} = \frac{0.2781}{0.6471} = 0.430$$

$$P(R=1 \mid W=1) = \frac{P(R=1, W=1)}{P(W=1)} =$$
$$\frac{\sum_{c,s} P(C=c, S=s, R=1, W=1)}{P(W=1)} = \frac{0.4851}{0.6471} = 0.708$$

Where,

$$P(W=1) = \sum_{c,r,s} P(C=c, S=s, R=r, W=1) = 0.6471$$

is a normalizing constant, equal to the probability (likelihood) of the data. So we see that it is more likely that the grass is wet because it is raining: the likelihood ratio is 0.7079/0.4298 = 1.647.

In the above example, notice that the two causes “competes" to "explain" the observed data. Hence S and R become conditionally dependent given that their common child, W, is observed, even though they are marginally independent. For example, suppose the grass is wet, but that we also know that it is raining. Then the posterior probability that the sprinkler is on goes down: P (S=1|W=1, R=1) = 0.1945, this is called "explaining away".

6.3.1.3 Top-Down and Bottom-Up Reasoning

In the water sprinkler example, we had evidence of an effect (wet grass), and inferred the most likely cause. This is called diagnostic, or "bottom up", reasoning, since it goes from effects to causes; it is a common task in expert systems. Bayes nets can also be used for causal, or "top down", reasoning. For example, we can compute the probability that the grass will be wet given that it is cloudy. Hence Bayes nets are often called "generative" models,

because they specify how causes generate effects. The particular aspect also termed as backward chaining can be exploited for setting the operating parameters as the desired out put is well known, thus in order to "effect" the desired out put the Bayesian tree can be used to find out the causes, this aspect can also be used for trouble shooting . In case the output obtained is not as per the desired specification one can find out the "cause".

6.4 Generation of Tree from RSP data.

In order to generate the tree the javabayes software has been used. When the software is loaded and run it generates two windows the javabayes editor and the javabayes console shown in Fig 6.2 below. In the console window is displayed a series of instructions as one proceeds to generate the network on the editor window. The opening instructions on the javabayes console are as follows, to start editing networks, press the Create button and click on the JavaBayes editor, or load a network using the file → Open menu. Thus it is highly user friendly and fully menu driven. The editor window has the following buttons for editing of the networks, create, move, delete, edit variable, edit function and edit networks. The create button is used to create the nodes, shown as green circles in the Fig 6.3 below, for this one has to just click the create button and then click any where inside the editor window, where one wants to create a node. Once all the nodes have been created as desired, the input nodes are connected to the output node by drawing arrows (arcs) from input to output using the mouse (just click and drag). Once the entire tree model is in place, if one wants to move or delete a node or arc, the same can be achieved using the move or delete buttons, for this one has to first click the move/delete button and then drag/click the desired node/arc which is to be moved/deleted.

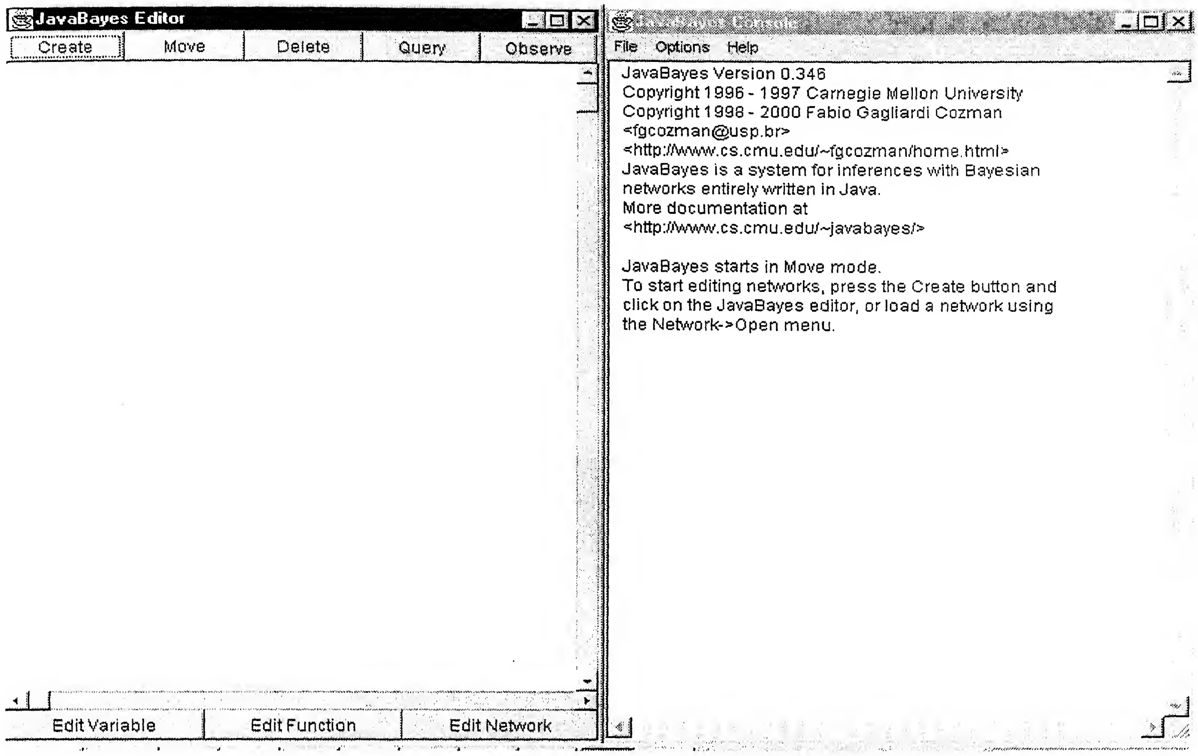


Fig6.2 JavaBayes Editor and Java Console

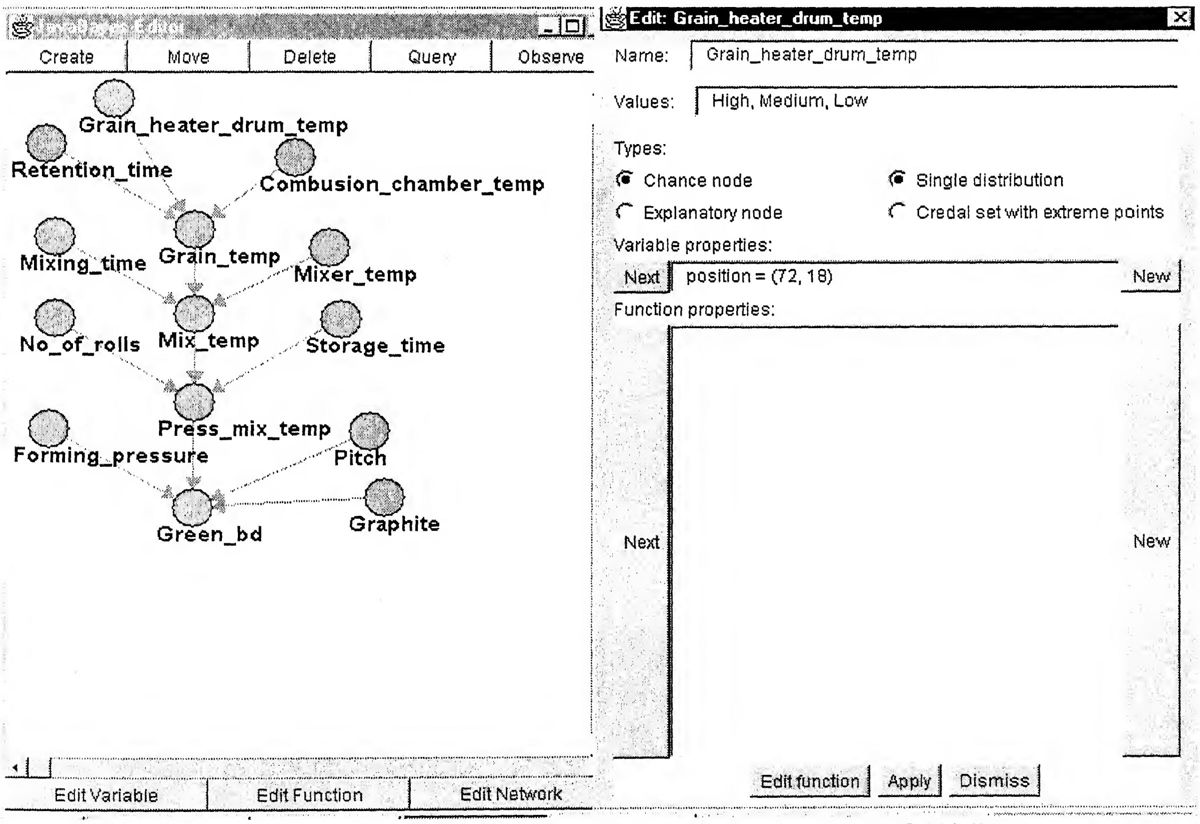


Fig 6.3 Tree Structure for GBD & Labeling Parameters for GHDT

Once the graph structure has been finalized, it is necessary to specify the parameters for the model. We must specify the Conditional Probability Distribution (CPD) at each node. This can be represented as a Conditional Probability table (CPT), which lists the probability that the child node takes on each of its different values for each combination of values of its parents. The operation is achieved using the edit variable and edit function buttons. Here again the edit variable button is clicked and then the variable to be edited is clicked which throws up a window as shown in the right side of Fig 6.3. From the window on the right side it can be seen that it allows you to change the name of the variable and the range of values that the variable can take as high, medium, low etc.

Once this is done for all the nodes of the tree one has to start entering the values for the Conditional Probability table (CPT). This done using the edit function button, when any node is clicked after clicking the edit function button it throws up a window as shown in Fig 6.4 below, which is for the node Grain Temperature.

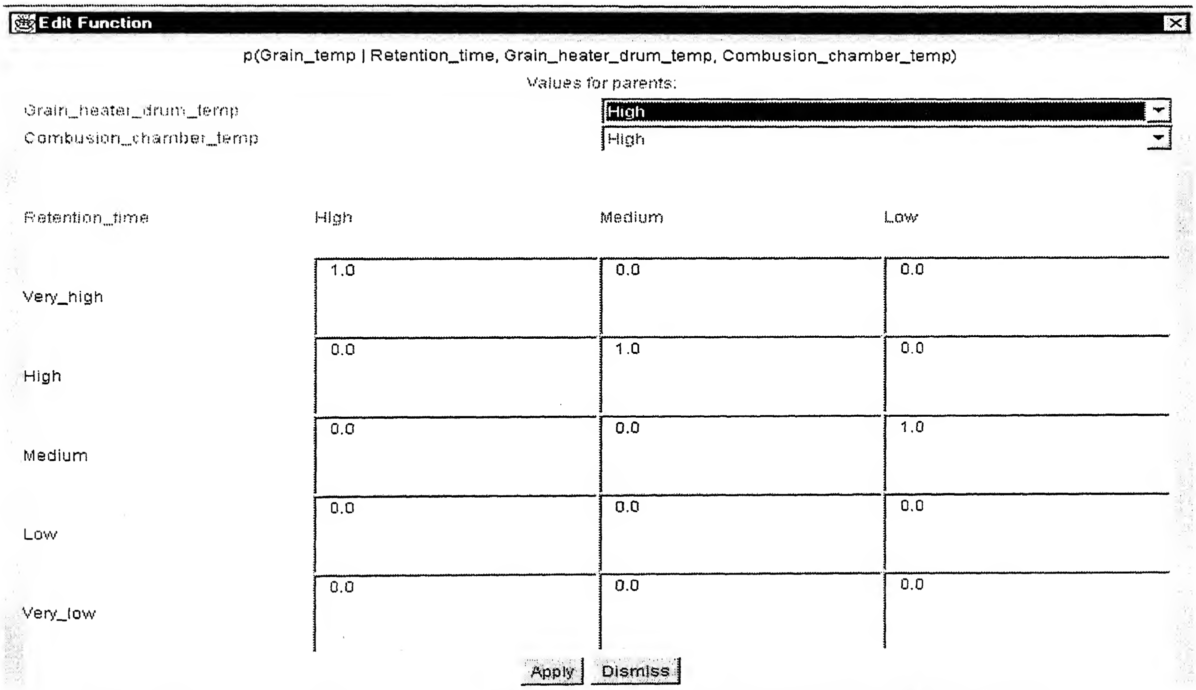


Fig 6.4 Edit Function Window for Entering the Conditional Probability Table.

The values entered here are those obtained from the rules generated by the experts, which is attached as per table 4.2 to 4.10 of Appendix A. Thus the tree has been developed only for the expert based opinion as the volume of data generation

been developed only for the expert based opinion as the volume of data generation for calculation of the probabilities was inadequate to develop a robust and meaningful system. To start tasting we have to use the query and observe buttons. The moment any of the nodes is clicked after clicking the query button, the probability distribution table of the nodes values gets listed out in the console window. Now if we observe that

Probability ("Combusion_chamber_temp") =1.0// P (Medium | evidence)

Probability ("Grain_heater_drum_temp") = 1.0// P (High | evidence)

Probability ("Retention time") =1.0// P (Low | evidence);

When we query for the node grain temperature we get,

Probability ("Grain_temp") = 1.0// P (Medium |evidence) shown in Fig 6.5

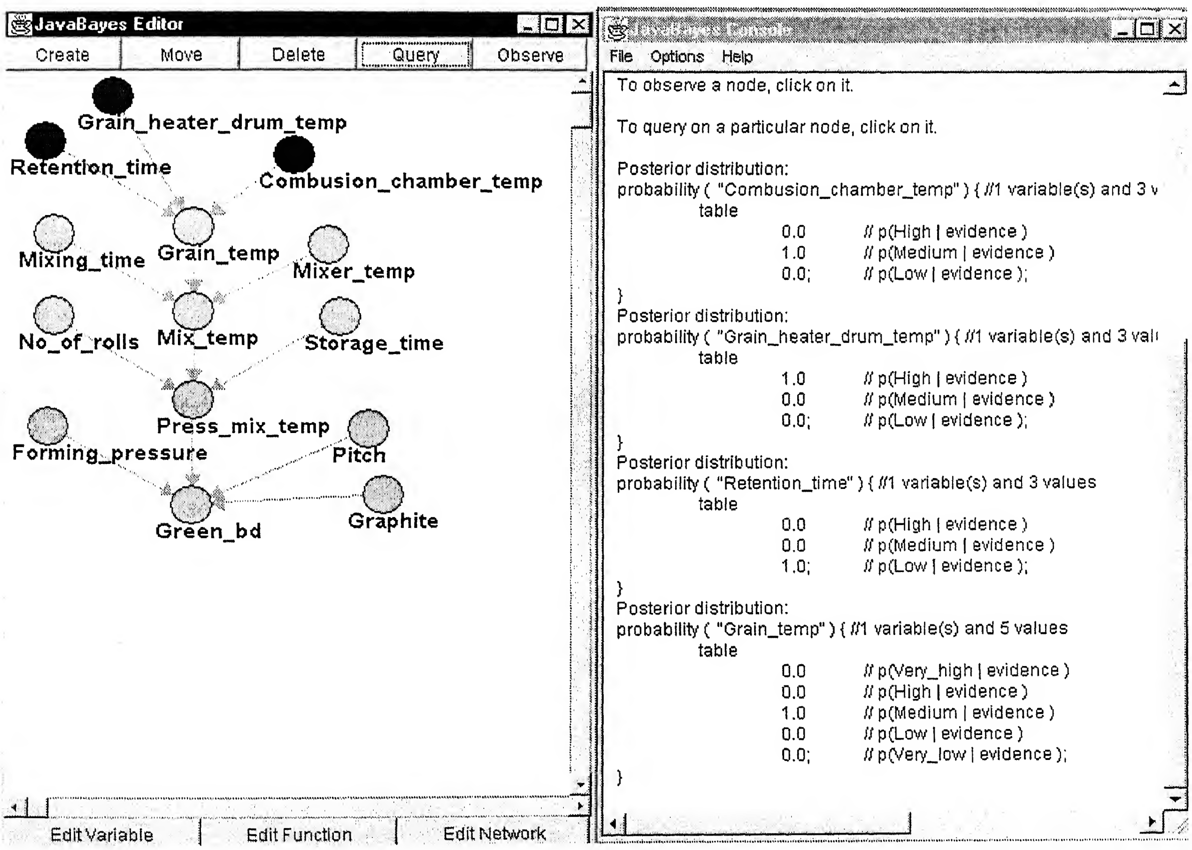


Fig 6.5 Probability Prediction for Grain Temp (Medium=1) Given RT, GHDT & CTI

6.5 Application of the Results Obtained for RSP

We proceed in this manner till we reach GBD, using "observe" and "query"

buttons to obtain results as compiled in Fig 6.6 to Fig 6.8 of Appendix F for a specific arbitrary example. For the processes following GBD an example has been dealt with and compiled as Fig 6.9 to 6.12 of the same Appendix F. Thus it can be seen that the effect of change in parameter in any of the parent node can be immediately queried at any level of the child node. Further, tree models have also been developed for backward chaining processes to diagnose the faults in case the out put shifts from the desired specification limits.

These Tree models developed for RSP can be used for the following purposes in the plant;

- It can be used for finding the effect of change in any single parameter on the final out put for example if want to know what the GBD is likely to be if GHDT alone is High, we just have to observe GHDT as High and query for GBD to get the result; Posterior distribution: probability ("Green_bd") {1 variable and 5 values table}

  0.03566529492455419 P (Very_high | evidence)

  0.13397347965249204 P (High | evidence)

  0.293278463648834 P (Medium | evidence)

  0.28321902149062644 P (Low | evidence)

  0.2538637402834934; P (Very_low | evidence); Hence the probability of the GBD being Medium or low is highest compared to the other three.
- Thus it can be used to refine the shape of the membership functions of the expert based system to match closely to the realistic data when ever the same is made available by the authorities for validation purposes.
- Validation of the rules from time to time as they are updated with additional data is most easily done using these tree structures as it is simple and easy to use even by the plant operator.
- During the validation of data the, these tree structures can be of great help to give an overall view of how the system is responding to the inputs.
- During the manufacturing process it can be readily used to predict the out put and hence take corrective action immediately.

- The trees developed for backward chaining can be used for trouble shooting/ fault diagnosis incase of wide variation in the output parameters.

CHAPTER 7
DEVELPOMENT OF FIS FROM EXPERT DATA

7.1 Introduction to Fuzzy Toolbox of Matlab

The Fuzzy Logic Toolbox is a collection of functions built on the MATLAB numeric computing environment. It provides tools for you to create and edit fuzzy inference systems within the framework of MATLAB. This toolbox relies heavily on graphical user interface (GUI) tools to help you accomplish your work, although you can work entirely from the command line if you prefer.

The toolbox provides two categories of tools:

•Command line functions

•Graphical, interactive tools

The first category of tools is made up of functions that you can call from the command line or from your own applications. Many of these functions are MATLAB M-files, series of MATLAB statements that implement specialized fuzzy logic algorithms. You can view the MATLAB code for these functions using the statement type function_name. You can change the way any toolbox function works by copying and renaming the M-file, then modifying your copy. You can also extend the toolbox by adding your own M-files.

Secondly, the toolbox provides a number of interactive tools that let you access many of the functions through a GUI. Together, the GUI- based tools provide an environment for fuzzy inference system design, analysis, and implementation.

7.2 Building Systems with the Fuzzy Logic Toolbox

Although it's possible to use the Fuzzy Logic Toolbox by working strictly from the command line, in general it's much easier to build a system graphically. There are five primary GUI tools for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox: the Fuzzy Inference System or **FIS Editor, the Membership Function Editor, the Rule Editor, the Rule Viewer, and the Surface Viewer**. These GUIs are dynamically linked, in that changes you make to the FIS using one of them, can affect what you see on any of the other open GUIs. You can have any or all of them open for any given system.

In addition to these five primary GUIs, the toolbox includes the graphical ANFIS Editor GUI, which is used for building and analyzing Sugeno-type adaptive neural fuzzy inference systems.

The **FIS Editor** handles the high level issues for the system: How many input and output variables? What are their names? The Fuzzy Logic Toolbox doesn't limit the number of inputs. However, the number of inputs may be limited by the available memory of your machine. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other GUI tools.

The **Membership Function Editor** is used to define the shapes of all the membership functions associated with each variable.

The **Rule Editor** is for editing the list of rules that defines the behavior of the system.

The **Rule Viewer and the Surface Viewer** are used for looking at, as opposed to editing, the FIS. They are strictly read-only tools. The Rule Viewer is a MATLAB-based display of the fuzzy inference diagram shown at the end of the last section of chapter 3. Used as a diagnostic, it can show (for example) which rules are active, or how individual membership function shapes are influencing the results. The Surface Viewer is used to display the dependency of one of the outputs on any one or two of the inputs—that is, it generates and plots an output surface map for the system.

7.3 FIS Developed For RSP

Two Fuzzy Inference Systems were developed for RSP, one based on the expert opinions where all rules were framed by taking inputs from the engineers and works at RSP and RDCIS, and the other was based on the data collected at the plant. In this chapter the former system will be discussed and the later will be discussed in chapter 8. The Expert based system was developed using fuzzy editor, while the data based system was developed using ANFIS editor.

The fuzzy inference engine developed for RSP using the tools mentioned in section 7.2 is shown in Fig 7.1 below. The five primary GUIs can all interact and exchange information. Any one of them can read and write both to the workspace and to the disk (the *read-only* viewers can still exchange plots with the workspace and/or the disk). For any fuzzy inference system, any or all of these five GUIs may be open. If more than one of these editors is open for a single system, the various GUI windows are aware of the existence of the others, and will, if necessary, update related windows. Thus if the names of the membership functions are changed using the Membership Function Editor, those changes are reflected in the rules shown in the Rule Editor. The editors for any number of different FIS systems may be open simultaneously. The FIS

Editor, the Membership Function Editor, and the Rule Editor can all read and modify the FIS data, but the Rule Viewer and the Surface Viewer do not modify the FIS data in any way.

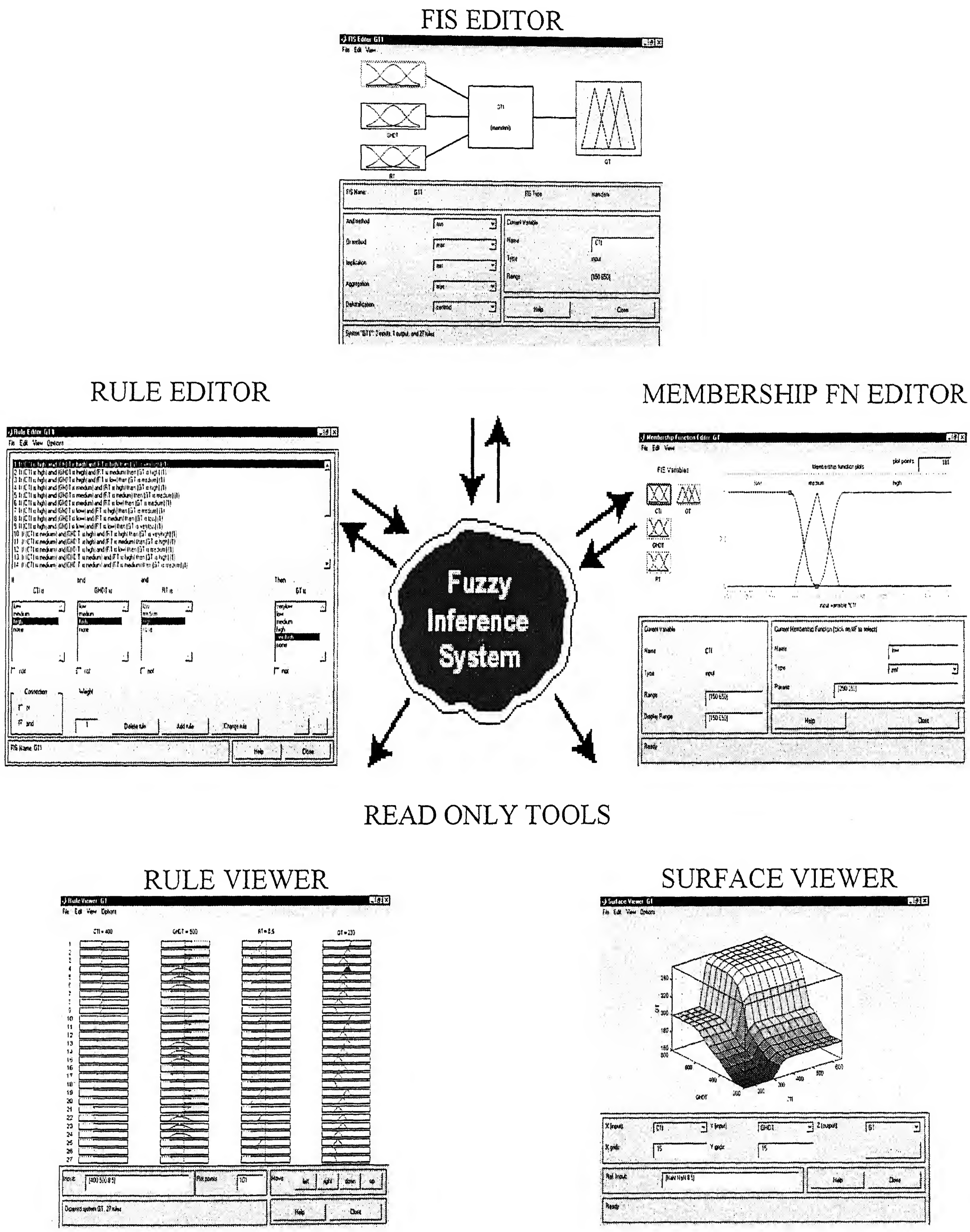


Fig 7.1 Fuzzy Inference System for RSP Showing the Five Tools In the GUI

For the purpose of developing the FIS model for RSP, the model of 9 sub processes mentioned in chapter4 table 4.1 was used.

7.3.1 Sub process 1

I shall describe this process with 3 inputs and one out put in detail. In this sub process the ranges for the input and output as given by the experts is given in table 7.1

Parameter	Range
CTI	150-650 degree centigrade
GHDT	200-800 degree centigrade
RT	2-15 minutes
GRAIN TEMPERATURE	90-300 degree centigrade

Table 7.1 Ranges for Parameters of Sub Processes - 1

The rules for the system are given in table 4.2 of Appendix A, the ranges for high, medium and low have been arrived at after a lot of deliberations between the various experts so as to obtain near true data results. Now we need to know the variation of the output, the shape of the curves, with respect to the three inputs, by plotting the curve for CTI Vs GT, GHDT Vs GT AND RT Vs GT we get the following plots given in Figures 7.2, 7.3 and 7.4. The remaining set of data are attached as Table 4.3 to 4.10 to Appendix A

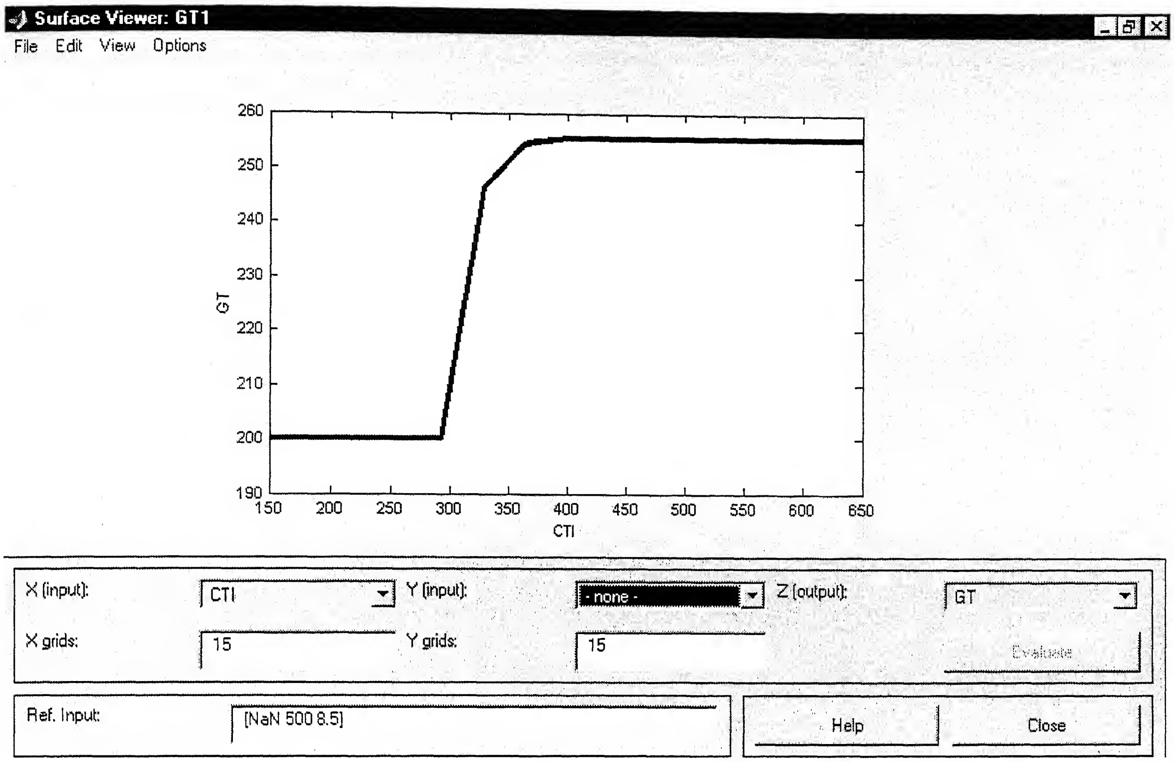


Fig 7.2 Plots for CTI Vs GT

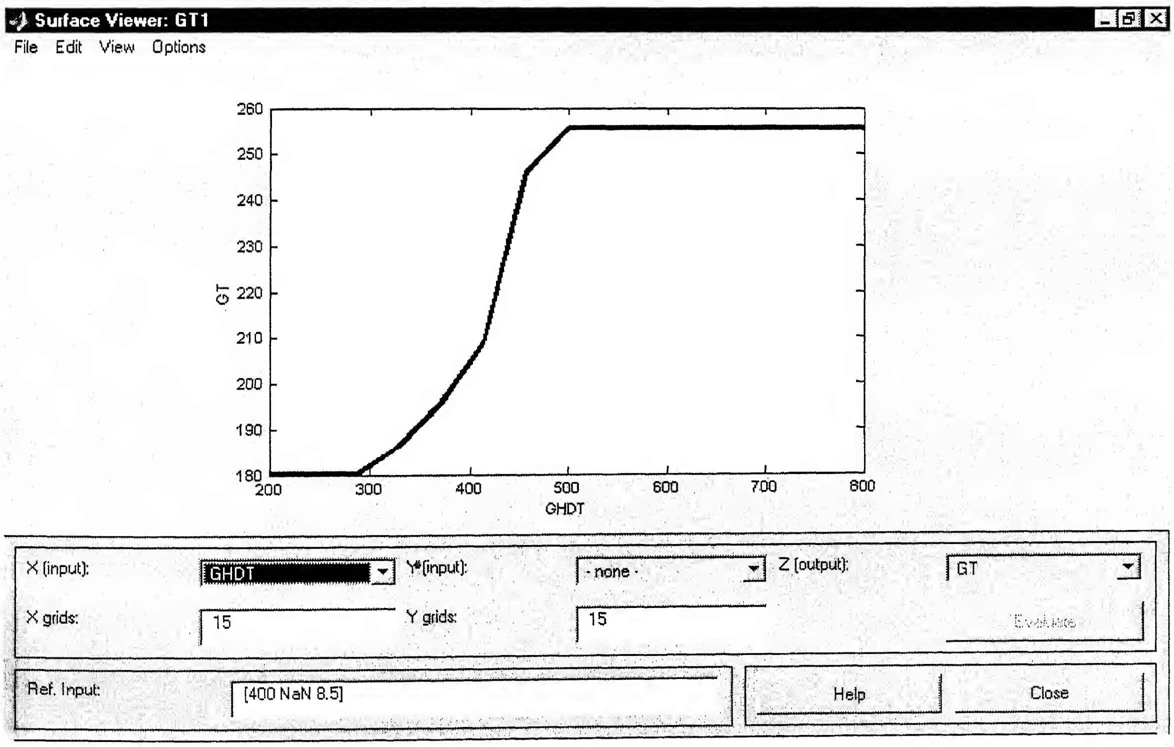


Fig 7.3 Plots for GHDT Vs GT

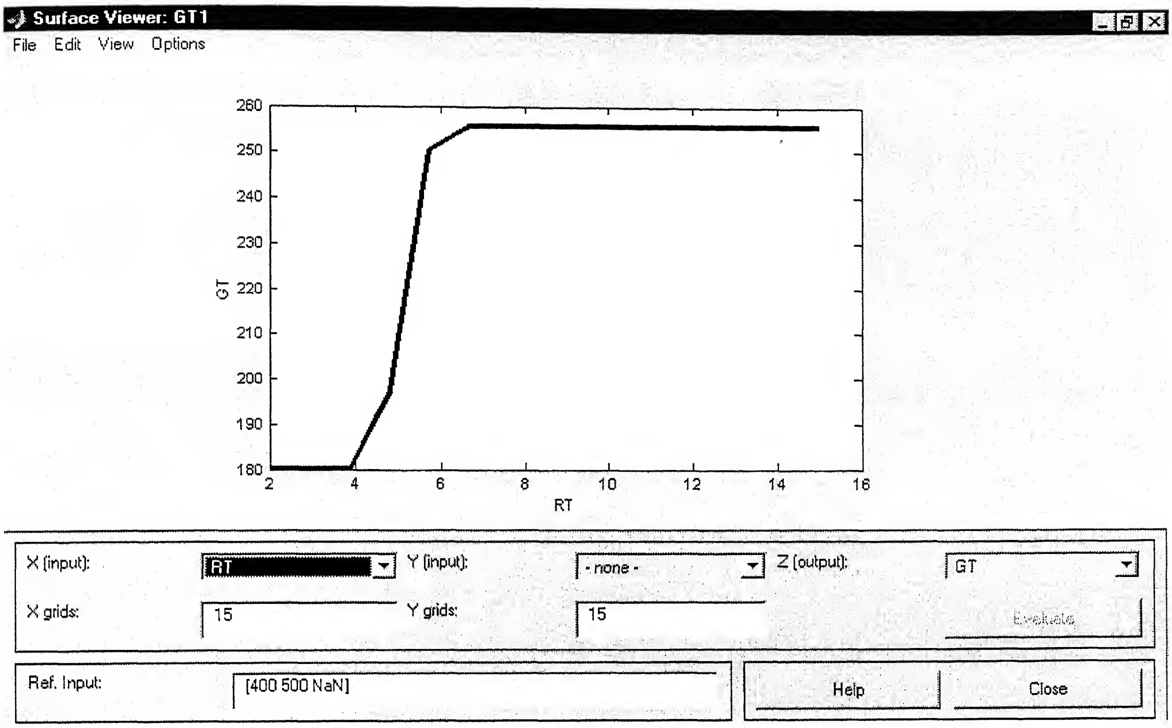


Fig 7.4 Plots for RT Vs GT

Now that we know the rules, and we know how the output varies with the inputs, let's start working with the GUI tools to construct a fuzzy inference system for the decision process of finding what the Grain Temperature is given what the three inputs are.

7.3.2 The Fis Editor

The FIS Editor displays general information about a fuzzy inference system. There's a simple diagram at the top that shows the names of each input variable on the left, and those of each output variable on the right. The sample membership functions shown in the boxes are just icons and do not depict the actual shapes of the membership functions.

Below the diagram is the name of the system and the type of inference used. The default, Mamdani-type inference, is what we've been describing so far. Another slightly different type of inference, called Sugeno-type inference, is also available. This method has already been explained in "Sugeno-Type Fuzzy Inference" in chapter 3 and which has been used in the ANFIS system to develop the data based FIS system explained.

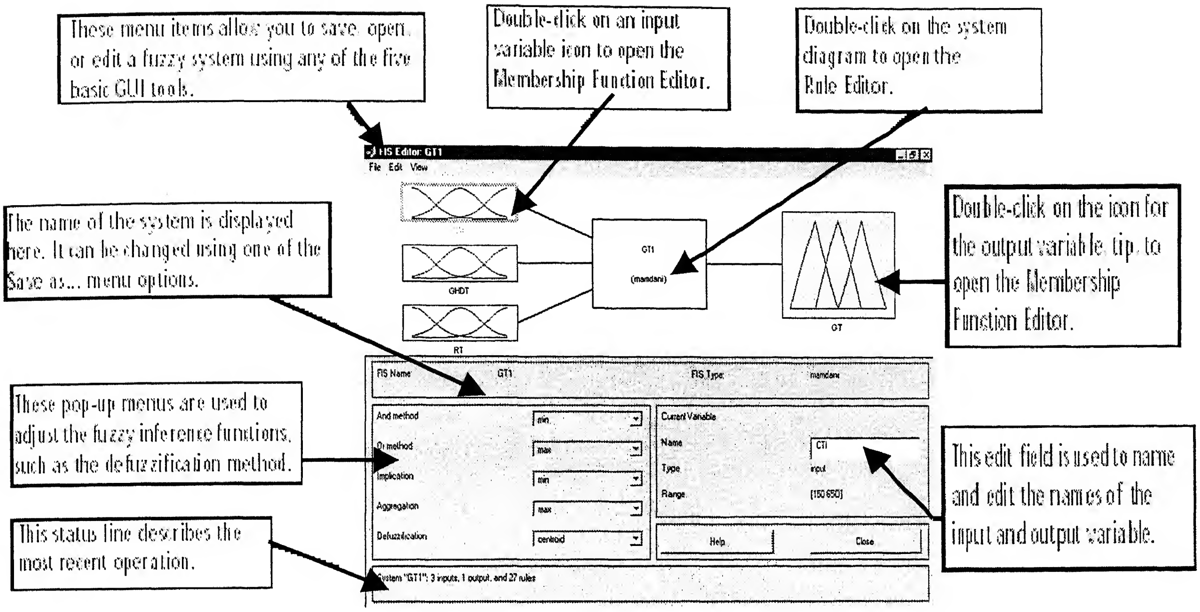


Fig 7.5 Fis Editor

In the fig 7.5 just below the name of the fuzzy inference system, on the left side of the figure, are the pop-up menus that allow you to modify the various pieces of the inference process. On the right side at the bottom of the figure is the area that displays the name of an input or output variable, its associated membership function type, and its range. The latter two fields are specified only after the membership functions have been. Below that region are the **Help** and **Close** buttons that call up online help and close the window, respectively. At the bottom is a status line that relays information about the system.

To start this system from scratch, type fuzzy at the MATLAB prompt. The generic untitled FIS Editor opens, with one input, labeled **input1**, and one output, labeled **output1**. For this sub system, we will construct a three-input, one output system, so go to the **Edit** menu and select **Add input** twice. A second and third yellow box labeled **input2 and input3** will appear. The three inputs we will have in our example are **service** and **food**. Our one output is **tip**. We'd like to change the variable names to reflect that, though:

- Click once on the left-hand (yellow) box marked **input1** (the box will be highlighted in red).
- In the white edit field on the right, change input1 to CTI and press **Return.**

➢ Click once on the left-hand (yellow) box marked **input2** (the box will be highlighted in red).

➢ In the white edit field on the right, change input2 to GHDT and similarly input3 to RT and press **Return**.

➢ Click once on the right-hand (blue) box marked **output1**.

➢ In the white edit field on the right, change output1 to GT.

7.3.3 Membership Function Editor

Next we have to define the membership functions associated with each of the variables. To do this, open the Membership Function Editor. You can open the Membership Function Editor in one of three ways:

- Pull down the **View** menu item and select **Edit Membership Functions....**
- Double-click on the icon for the output variable, **GT**.
- Type mfedit at the command line.

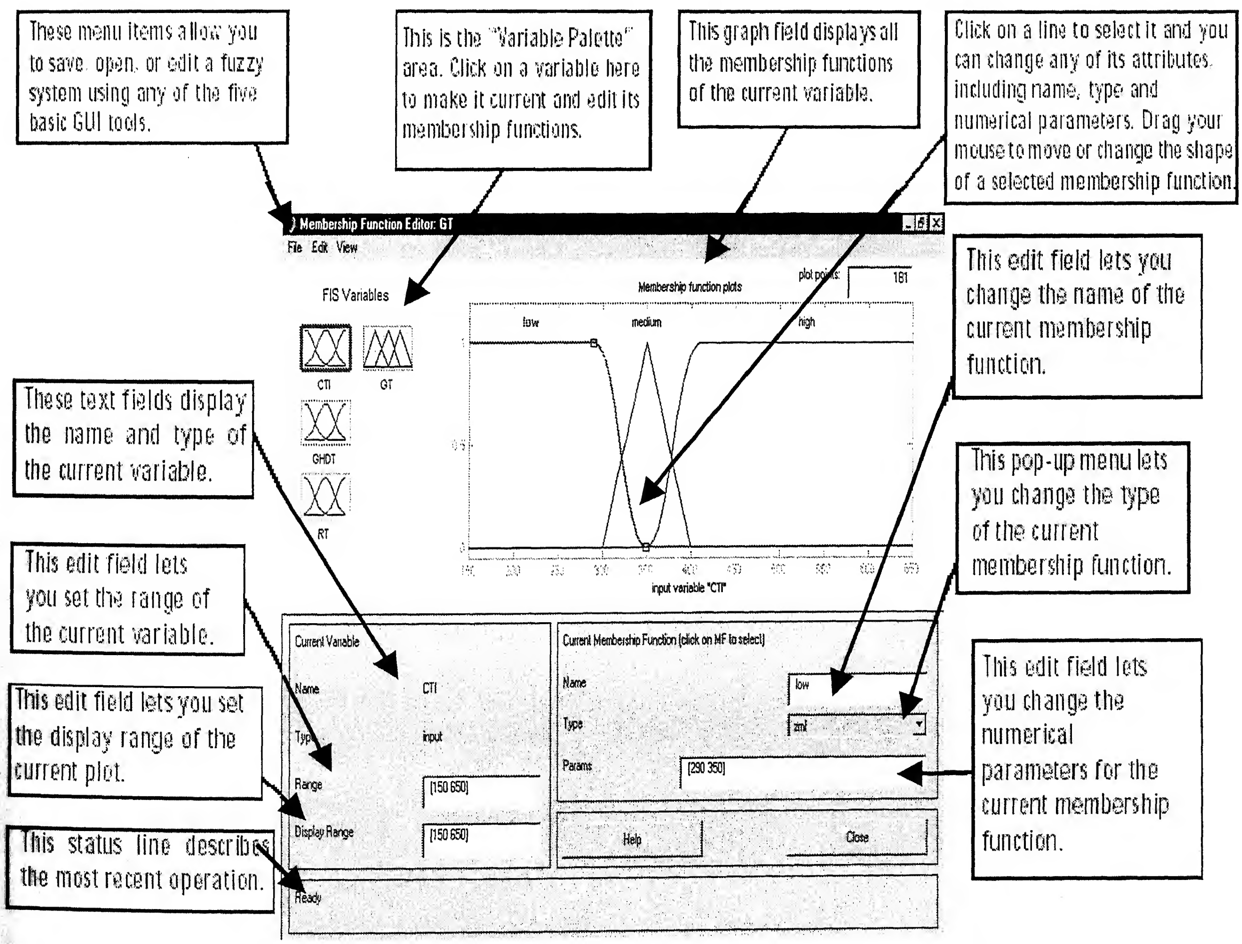


Fig 7.6 Membership Function Editor

The Membership Function Editor in fig 7.6 shares some features with the FIS Editor. In fact, all of the five basic GUI tools have similar menu options, status lines, and **Help** and **Close** buttons. The Membership Function Editor is the tool that lets you display and edits all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system.

On the upper left side of the graph area in the Membership Function Editor is a "Variable Palette" that lets you set the membership functions for a given variable. To set up your membership functions associated with an input or an output variable for the FIS, select an FIS variable in this region by clicking on it.

Next select the **Edit** pull-down menu, and choose **Add MFs**. A new window will appear which allows you to select both the membership function type and the number of membership functions associated with the selected variable. The built in membership functions available in the tool box are **triangular, trapezoidal, gbell, Gaussian, sigma, dsigma, psigma, pi, S and Z membership functions**. In the lower right corner of the window are the controls that let you change the name, type, and parameters (shape), of the membership function, once it has been selected. The membership functions from the current variable are displayed in the main graph. These membership functions can be manipulated in two ways. You can first use the mouse to select a particular membership function associated with a given variable quality, (such as high, for the variable, CTI), and then drag the membership function from side to side. This will affect the mathematical description of the quality associated with that membership function for a given variable. The selected membership function can also be tagged for dilation or contraction by clicking on the small square drag points on the membership function, and then dragging the function with the mouse toward the *outside,* for dilation, or toward the *inside*, for contraction. This will change the parameters associated with that membership function.

Below the Variable Palette is some information about the type and name of the current variable. There is a text field in this region that lets you change the limits of the current variable's range (universe of discourse) and another that lets you set the limits of the current plot (which has no real effect on the system). The process of specifying the input membership functions for this two input tipper problem is as follows:

- Select the input variable, **CTI**, by double-clicking on it. Set both the **Range** and the **Display Range** to the vector [150 650].
- Select **Add MFs.** from the **Edit** menu. The window below pops open.
- Use the pull-down tab to choose **trimf** for **MF Type** and **3** for **Number of MFs**. This adds three triangle MFs to the input variable CTI.
- Click once on the leftmost triangle. Change the name of the curve to low. To adjust the shape of the membership function, either use the mouse, as described above, or type in a desired parameter change, and then click on the membership function. The parameter listing for this curve is [300 350]
- Name middle triangle, medium, and the rightmost triangle, high. Reset the associated parameters to [300 350 400] and [350 400].
- Select the input variable, **GHDT**, by clicking on it. Set both the **Range** and the **Display Range** to the vector [200 800].
- Select **Add MFs.** from the **Edit** menu and add three **trimf** curves to the input variable GHDT.
- Similarly name the triangles as low, medium and high with parameters [300 400], [300 400 500] and [400 500]
- Select the input variable, **RT**, by clicking on it. Set both the **Range** and the **Display Range** to the vector [2 15].
- Similarly name the triangles as low, medium and high with parameters [4 5], [4 5 6] and [5 6]
- Select the output variable, **GT**, by clicking on it. Set both the **Range** and the **Display Range** to the vector [90 300].
- Select **Add MFs.** from the **Edit** menu and add five **trimf** curves to the output variable GT.
- Similarly name the triangles as very low, medium high and very high with parameters [140 160], [140 160 180], [160 180 200], [180 200 220] and [200 220]

Now the final system should look something like the fig 7.7 below,

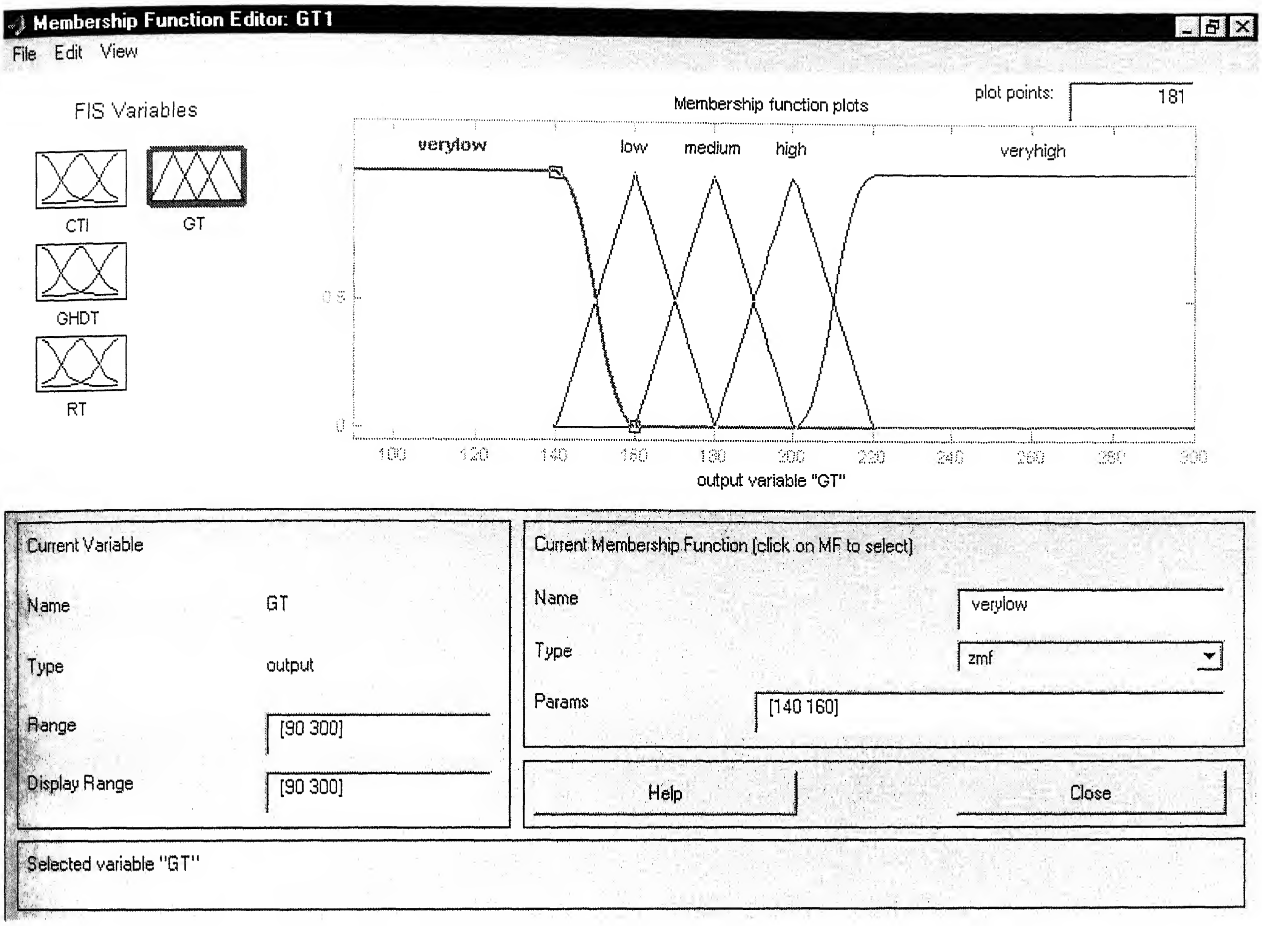


Fig 7.7 Final FIS System for Grain Temperature (GT)

7.3.4 Rule Editor

Now one is ready to write down the rules as give in the table 7.2. To call up the Rule Editor, go to the **View** menu and select **Edit rules**, or type ruleedit at the command line.

Constructing rules using the graphical Rule Editor interface as shown in fig 7.8 below is fairly self-evident. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows you to construct the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item. Choosing **none** as one of the variable qualities will exclude that variable from a given rule. Choosing **not** under any variable name will negate the associated quality. Rules may be changed, deleted, or added, by clicking on the appropriate button.

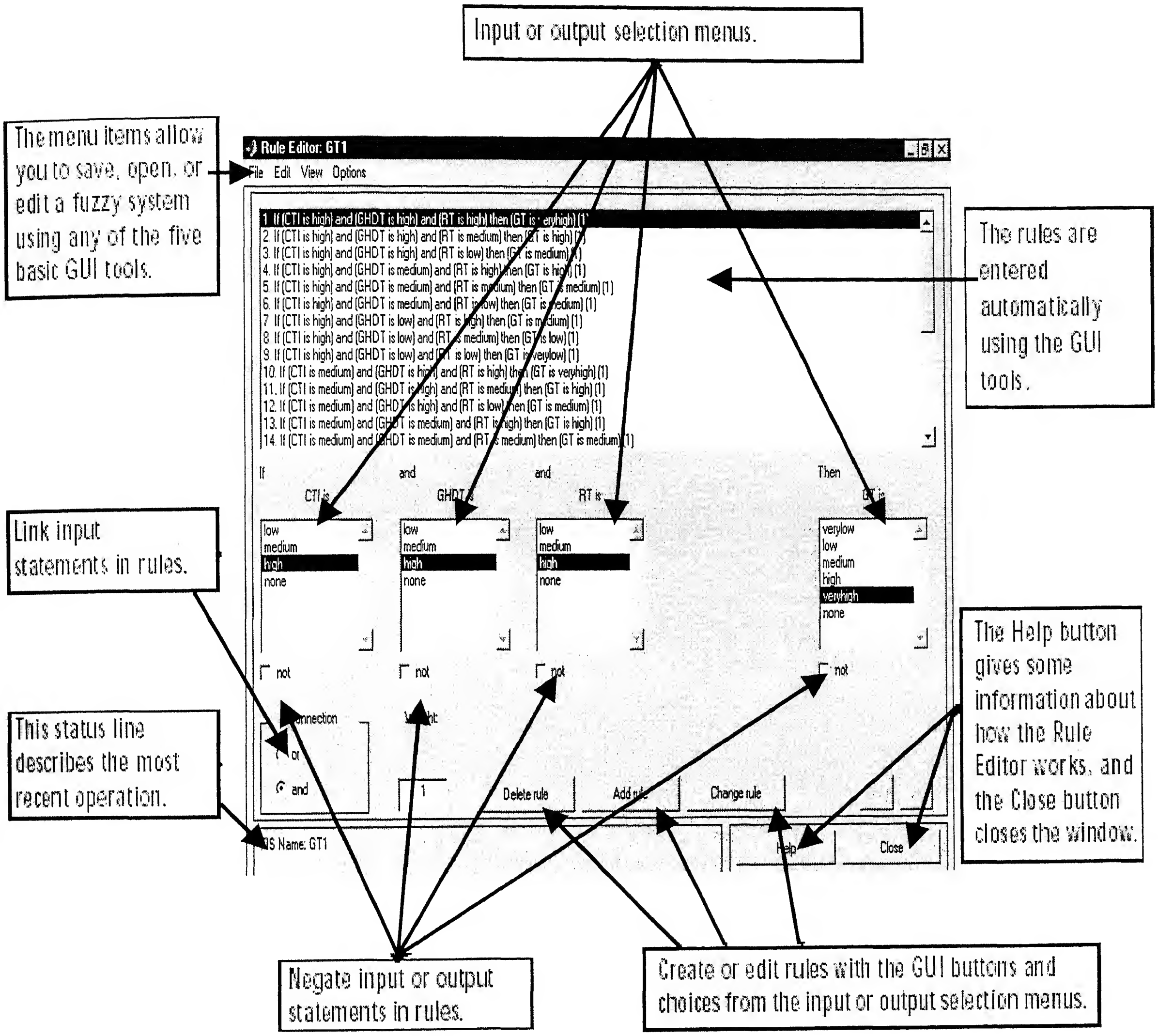


Fig 7.8 Rule Editor

The Rule Editor also has some familiar landmarks, similar to those in the FIS Editor and the Membership Function Editor, including the menu bar and the status line. The **Format** pop-up menu is available from the **Options** pull-down menu from the top menu bar; this is used to set the format for the display. Now one has to start entering the rules as per the table 7.2 by selecting the entries from the boxes. Once all the 27 rules have been entered the GT.fis system is fully ready for viewing the rules and surfaces, which can be done using the view menu.

7.3.5 Rule Viewer and Surface Viewer.

On selecting the rule viewer the fig7.9 pops up as under,

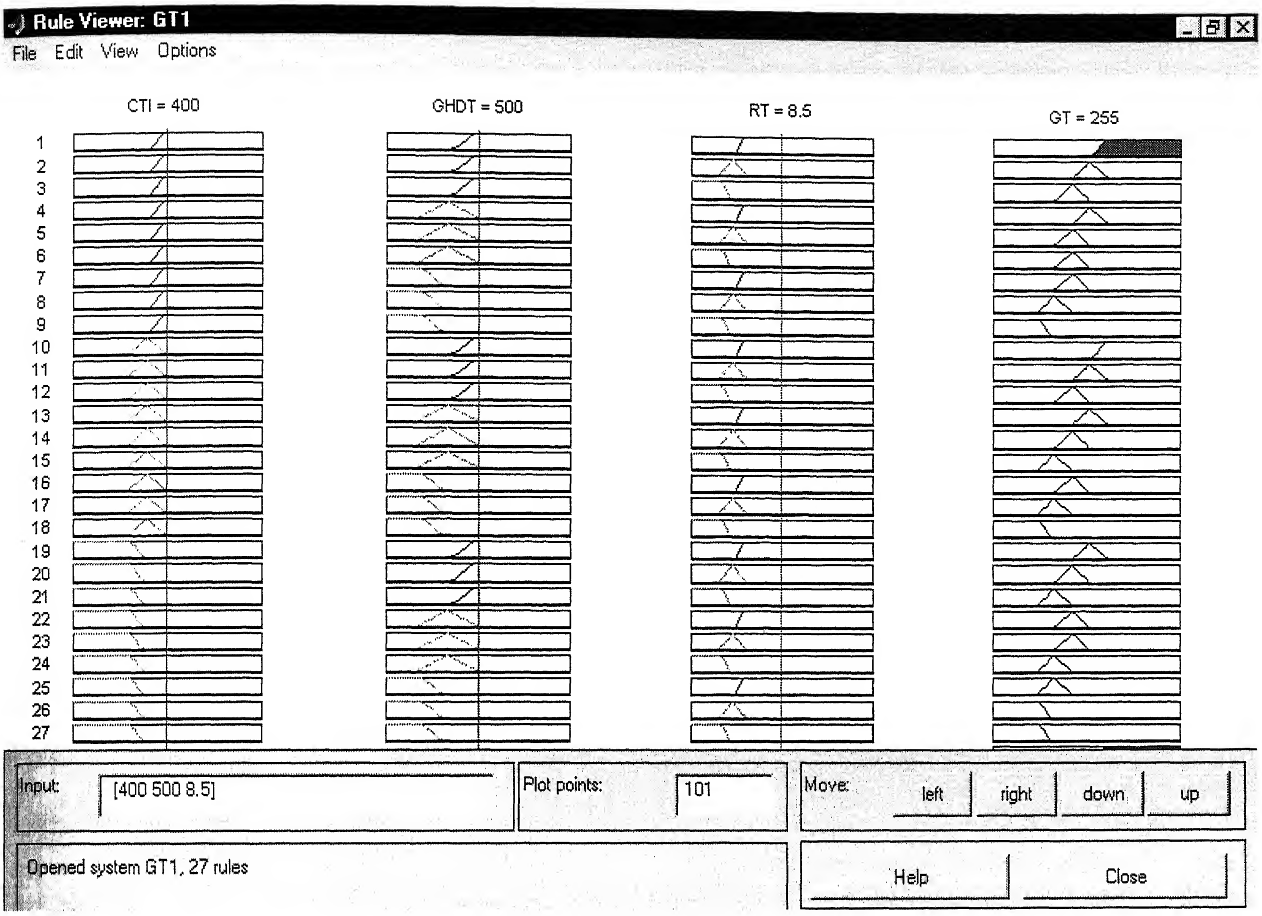


Fig 7.9 Rule Viewer

The Rule Viewer allows you to interpret the entire fuzzy inference process at once. The Rule Viewer also shows how the shape of certain membership functions influences the overall result. Since it plots every part of every rule, it can become unwieldy for particularly large systems, but, for a relatively small number of inputs and outputs, it performs well (depending on how much screen space you devote to it) with up to 30 rules and as many as 6 or 7 variables.

The Rule Viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system. If you want to see the entire output surface of your system, that is, the entire span of the output set based on the entire span of the input set, you need to open up the Surface Viewer. This is the last of the five basic GUI tools in the Fuzzy Logic Toolbox, and you open it by selecting **View surface...** from the **View** menu. The following screen pops up as in fig 7.10 below.

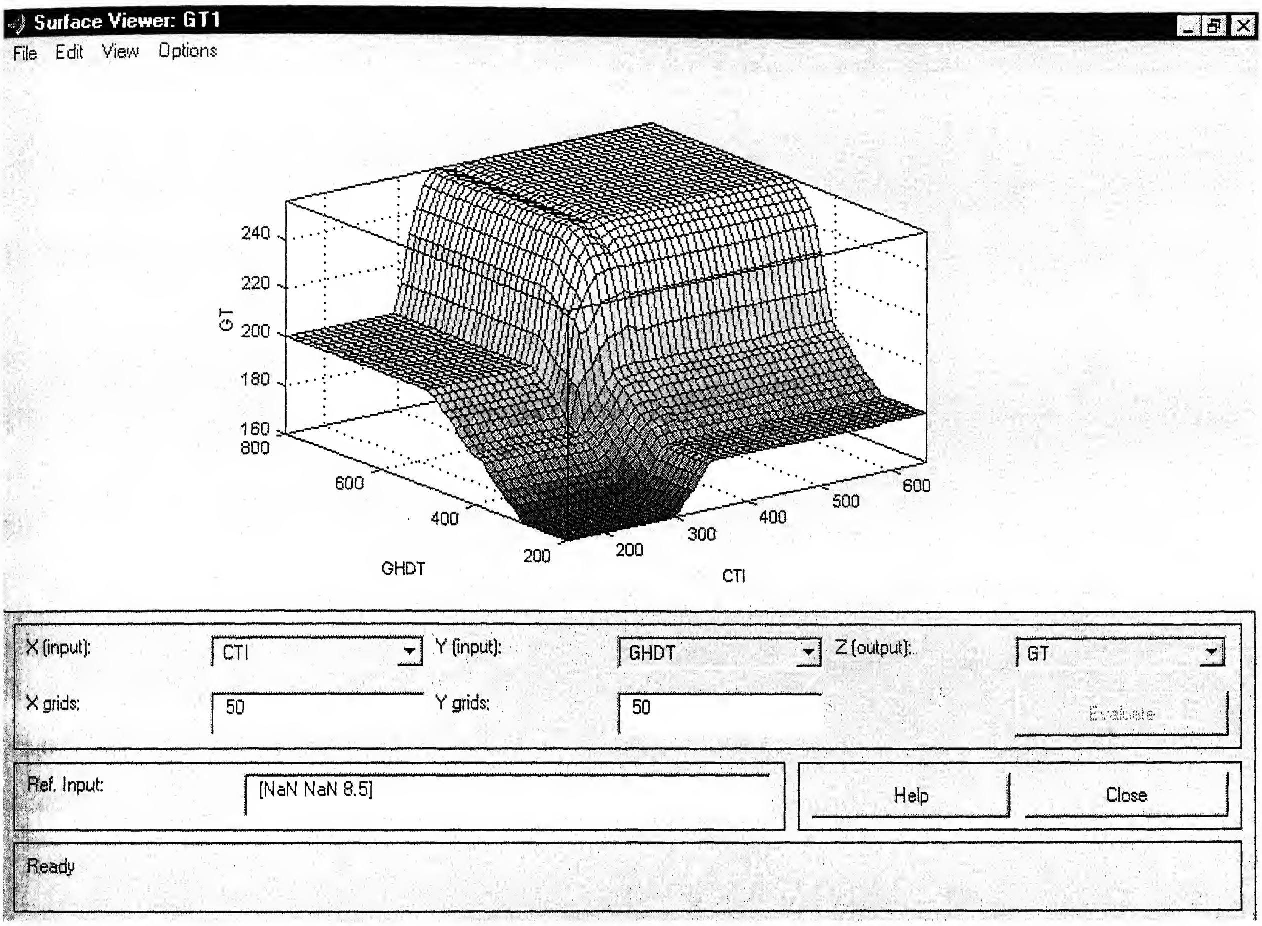


Fig 7.10 Surface Viewer Plot for CTI & GHDT Vs GT

Upon opening the Surface Viewer, we are presented with three-dimensional plots that MATLAB can adeptly manage. When we move beyond three dimensions as in this case overall, we start to encounter trouble displaying the results. Accordingly, the Surface Viewer is equipped with pop-up menus that let you select any two inputs and any one output for plotting. Just below the pop-up menus are two text input fields that let you determine how many *x-axis* and *y-axis* grid lines you want to include. This allows you to keep the calculation time reasonable for complex problems. Pushing the **Evaluate** button initiates the calculation, and the plot comes up soon after the calculation is complete. To change the *x-axis* or *y-axis* grid after the surface is in view, simply change the appropriate text field, and click on either **X-grids** or **Y-grids**, according to which text field you changed, to redraw the plot.

7.4 Final Output and Results

Similarly the remaining sub systems were developed as MIX_TEMP, PMT, GBD, COKEDPROCITY, CCS, HMOR, SCR and OXYDATION RESISTANCE. The membership functions for each of these are attached as per Fig 7.11 to 7.18 of Appendix G. The rule sets for each of these are attached as Table 4.3 to Table 4.10 in Appendix A. The surface plots up to the Green BD stage are attached as per Fig 7.19 to 7.39 of Appendix H. The first 4 sub systems were combined using a matlab programme file called GBD_LONG.m. This programme accepts input from files and the out of the first is fed to the next along with additional inputs and this chain goes on till we get the final out as Green Bulk Density. Individual matlab programme files have also been made to accept inputs in vector form. In addition programmes have also been made so that the user can input the data file from the keyboard when prompted to do so. The results obtained using sample data are attached as per tables 7.11 to 7.15 of Appendix I.

Validation of the expert based system requires a large volume of low noise data so that the membership functions can be modified to achieve the desired results. The non availability such a data set has meant that this has to be vigorously followed up in the future and the Expert based Fuzzy Inference system should be fine tuned to achieve best results.

CHAPTER 8

DEVELOPMENT OF FIS FORM PLANT DATA

8.1 Overview of ANFIS SYSTEM

8.1.1Sugeno-Type Fuzzy Inference

The fuzzy inference process we've been referring to so far is known as Mamdani's fuzzy inference method. It's the most commonly seen fuzzy methodology. In this section we discuss the Sugeno method of fuzzy inference as the ANFIS EDITOR of matlab toolbox utilizes this method only. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani-type of fuzzy inference and Sugeno-type is that the output membership functions are only linear or constant for Sugeno-type fuzzy inference.

A typical fuzzy rule in a *zero-order Sugeno fuzzy model* which has the form if x is A and y is B then $z = k$, where A and B are fuzzy sets in the antecedent, while k is a crisply defined constant in the consequent. When the output of each rule is a constant like this, the similarity with Mamdani's method is striking. The only distinctions are the fact that all output membership functions are singleton spikes, and the implication and aggregation methods are fixed and can not be edited. The implication method is simply multiplication, and the aggregation operator just includes all of the singletons.

- **Advantages of the Sugeno Method**
  - It's computationally efficient.
  - It works well with linear techniques (e.g., PID control).
  - It works well with optimization and adaptive techniques.
  - It has guaranteed continuity of the output surface.
  - It's well-suited to mathematical analysis.
- **Advantages of the Mamdani Method**
  - It's intuitive.
  - It has widespread acceptance.
  - It's well-suited to human input.

8.2 Anfis and the ANFIS Editor GUI

In this section we discuss the use of the function anfis and the ANFIS Editor GUI in the Fuzzy Logic Toolbox. These tools apply fuzzy inference techniques to data modeling. As you have seen from the other fuzzy inference GUIs, the shape of the membership functions depends on parameters, and changing these parameters will change the shape of the membership function.

Instead of just looking at the data to choose the membership function parameters, we will see how membership function parameters can be chosen automatically using these Fuzzy Logic Toolbox applications.

8.2.1 Data Modeling

For modeling the manufacturing process of RSP we were in the initial stages given a data set of 86 observations consisting of 10 inputs and 1 output, the same is in table 4.13 attached as Appendix C. An additional set of 31 data sets (in table 4.12 of Appendix C) was sent subsequently making a total of 117 observations hence the modeling was done for both the 86 and 117 data sets separately. Since the membership functions chosen for the various attributes in the expert system developed by expert advice was to certain extent arbitrary. Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen so as to tailor the membership functions to the input/output data in order to account for these types of variations in the data values. This is where the *neuro-adaptive* learning techniques incorporated into anfis in the Fuzzy Logic Toolbox can help. The inputs and output for RSP for which data was made available is as under,

- Inputs
  - Combustion chamber temperature
  - Grain heater drum temperature
  - Grain retention time
  - Mixer temperature
  - Mixing time
  - Mix temperature
  - Storage time
  - Rolls
  - Press mix temperature
  - Forming pressure
- Outputs
  - Green bulk density (GBD)

8.2.2 Learning and Inference through ANFIS

The basic idea behind these neuro-adaptive learning techniques is that these techniques provide a method for the fuzzy modeling procedure to *learn* information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks. The Fuzzy Logic Toolbox function that accomplishes this membership function parameter adjustment is called anfis (*adaptive neuro-fuzzy inference System)*. Using a given input/output data set, the toolbox function anfis constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back propagation algorithm alone, or in combination with a least squares type of method. This allows your fuzzy systems to learn from the data they are modeling.

8.2.3 Importance of Accurate Data for Good Modeling

The modeling approach used by anfis is similar to many system identification techniques. First, you hypothesize a parameterized model structure (relating inputs to membership functions to rules to outputs to membership functions, and so on). Next, you collect input/output data in a form that will be usable by anfis for training. You can then use anfis to *train* the FIS model to emulate the training data presented to it by modifying the membership function parameters according to a chosen error criterion. In general, this type of modeling works well if the training data presented to anfis for training (estimating) membership function parameters is fully representative of the features of the data that the trained FIS is intended to model. This is not always the case as in the data given to us by RSP has only 117 observations where we would require about 3000 to 4000 data sets. More over the data collected had noisy measurements, and hence the training data could be representative of all the features of the data that will be presented to the model.

8.2.4 Model Validation Using Testing Data and Checking Data Sets

Model validation is the process by which the input vectors from input/output data sets on which the FIS was not trained, are presented to the trained FIS model, to see how well the FIS model predicts the corresponding data set output values. This is accomplished with the ANFIS Editor GUI using the *testing data set*. In this case the 31 data set received subsequently was used as testing data set. The basic idea behind using a

checking data set for model validation is that after a certain point in the training, the model begins over fitting the training data set. In principle, the model error for the checking data set tends to decrease as the training takes place up to the point that over fitting begins, and then the model error for the checking data suddenly increases.

8.2.5 The ANFIS Editor GUI

To get started with the ANFIS Editor GUI, type anfisedit on the command window and the following window as in fig8.1 will pop up.

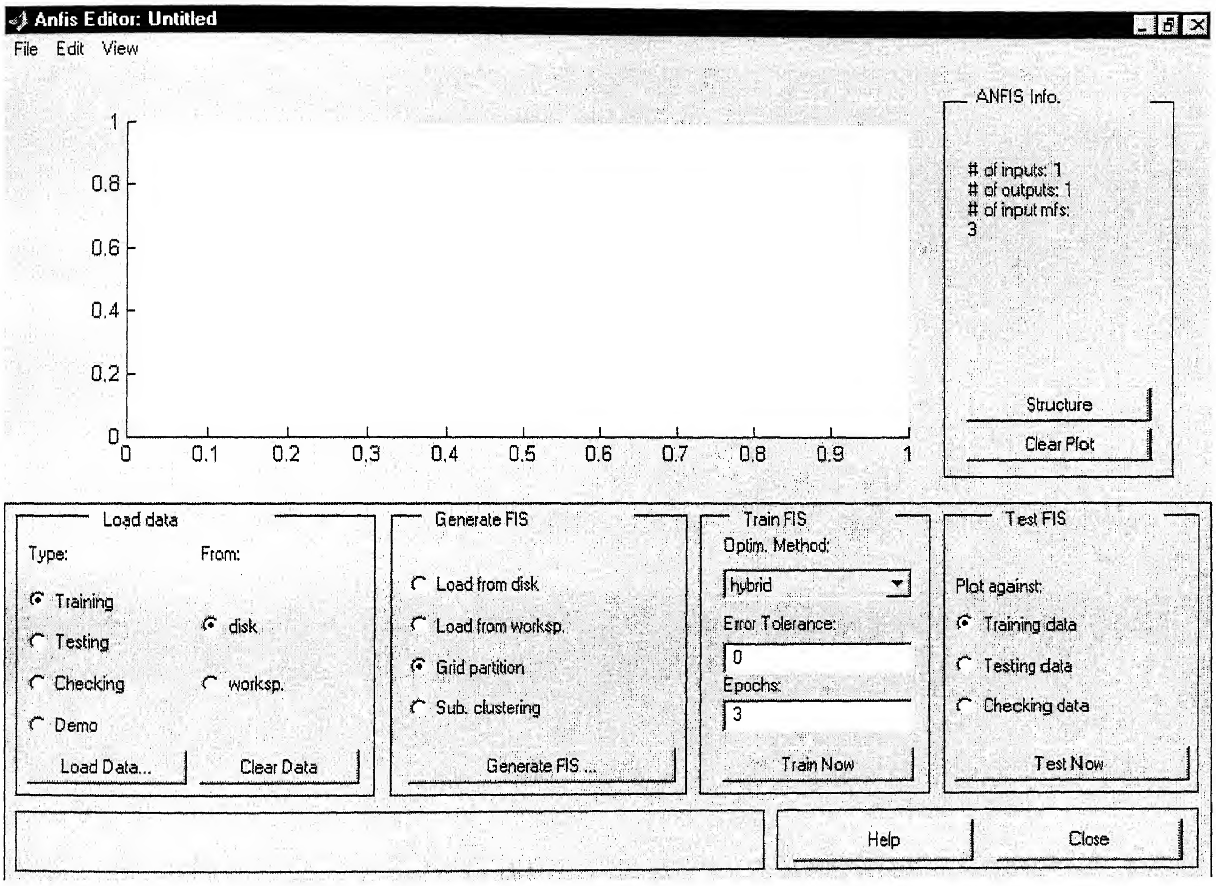


FIG 8.1 Anfis GUI

8.2.5.1 Loading Data

From the GUI above one can, Load data (training and testing, and checking) by selecting appropriate radio buttons in the **Load data** portion of the GUI and then selecting **Load Data** the loaded data is plotted on the plot region. Here we load the 86 data sets for training and 31 data sets for testing. Since the numbers of data sets were not

sufficient checking data could not be loaded. This is how it looks once the data has been loaded as in fig 8.2

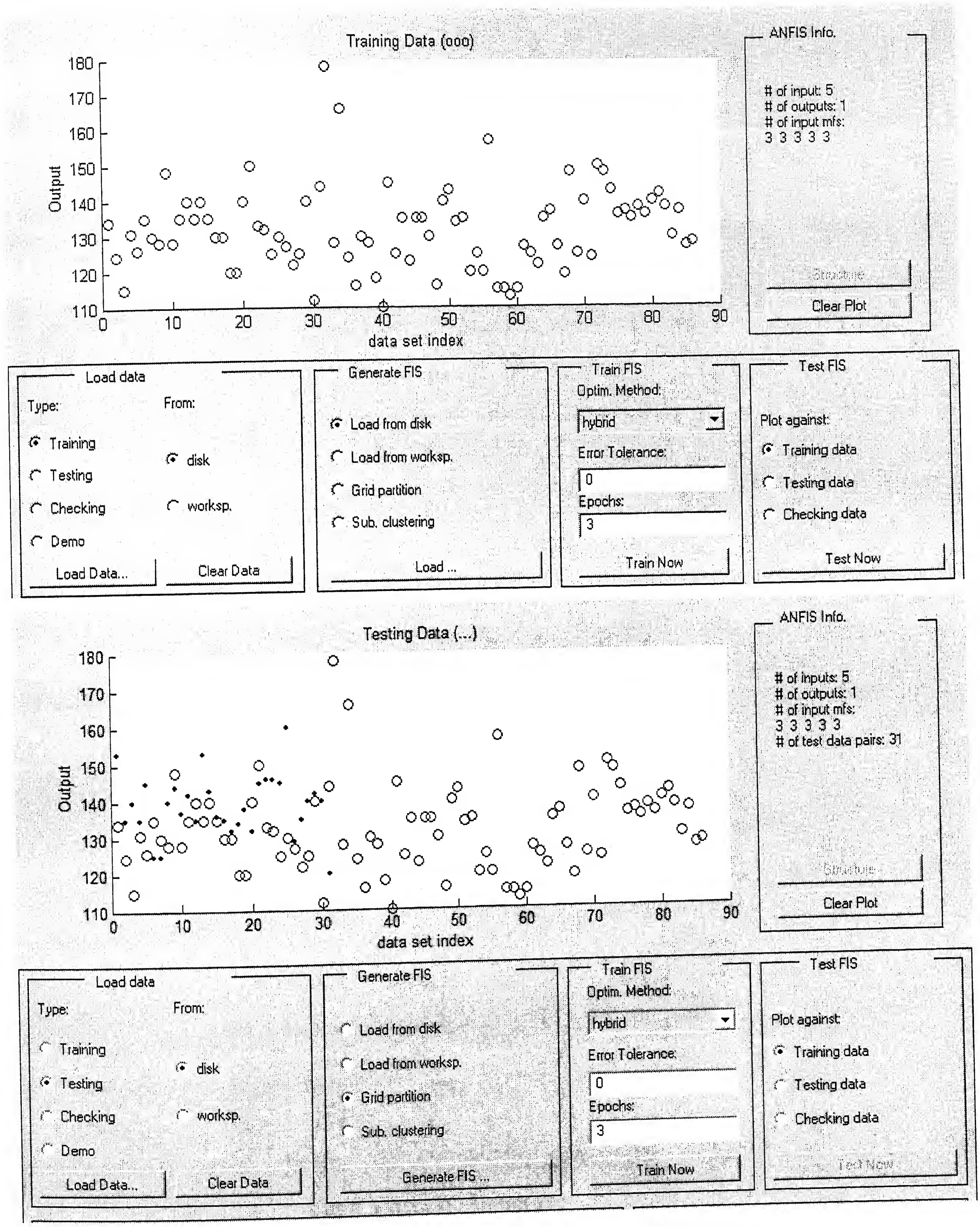


Fig 8.2 Loading of Training and Testing Data

8.2.5.2 Generating the FIS Model

Generate an initial FIS model or load an initial FIS model using the options in the **Generate FIS** portion of the GUI.

- Here we generate two Fis models using Grid partitioning option where we specify 3 MFs for each input and 3 MFs for the outputs. The two Fis is for the data sets broken up into two parts the first having 5 input and 1 out put and the second having the output of the first as one of its input along with remaining 4 of the 10 inputs. This splitting of the inputs had to be done because with 10 inputs taken together and each with 3 MFs, the system would have to generate 3^{10} rules and hence was running out of memory. Further the membership functions selected were triangular and trapezoidal only, because for all other memberships' functions (gbell, **Gaussian, sigma, dsigma, psigma, pi, S and Z membership functions)** the results were not very encouraging. Hence two sets of Fis were generated, one set for triangular membership functions and the other for trapezoidal.

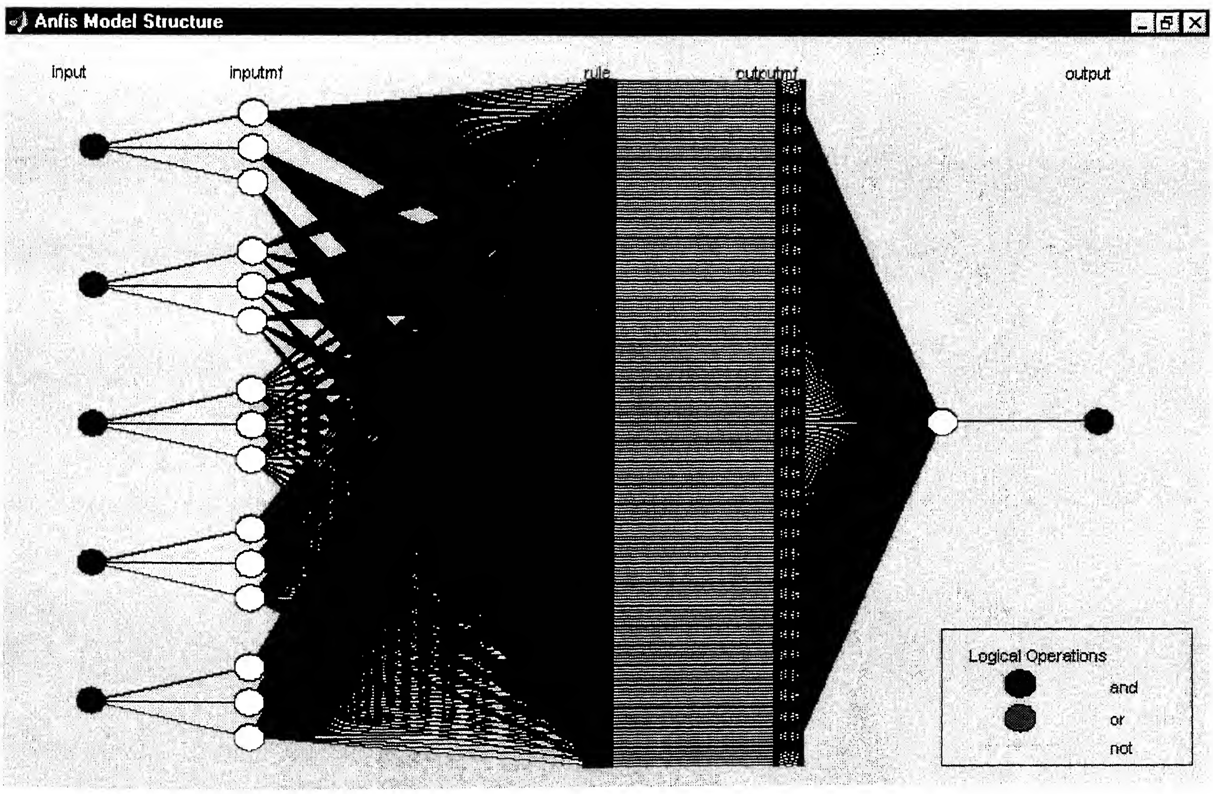


Fig 8.3 ANFIS Structure

➢ View the FIS model structure once an initial FIS has been generated or loaded by selecting the **Structure** button. This is how the structure looks like as in fig 8.3 above

8.2.5.3 Training the Generated FIS

➢ Train the Fis model using the parameter optimization method as Hybrid (mixture of back propagation and least squares) and number of epochs as 300 and margin of error as 0.01 and click train now. As the training progresses the following screen is displayed as per fig 8.4 below. And once training is completed the screen displayed is shown in fig 8.5.

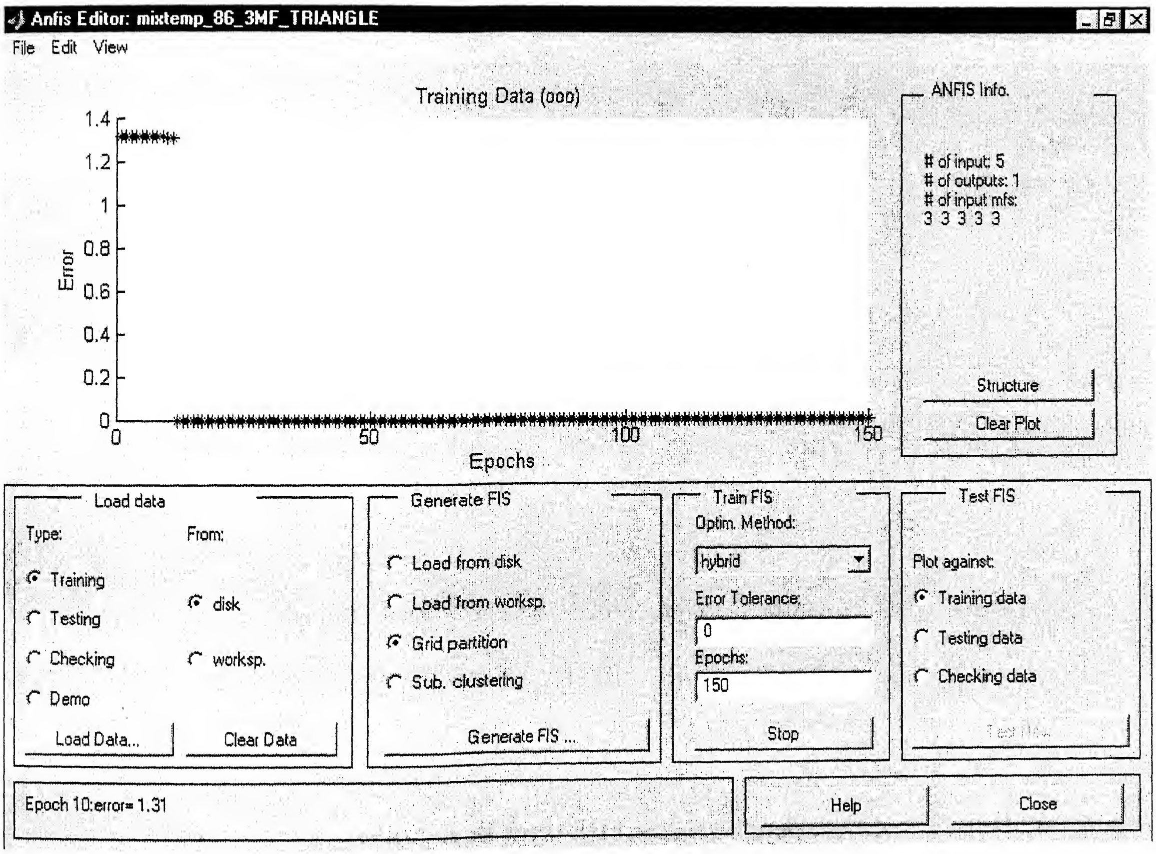


Fig 8.4 Training of Data for Generation of FIS

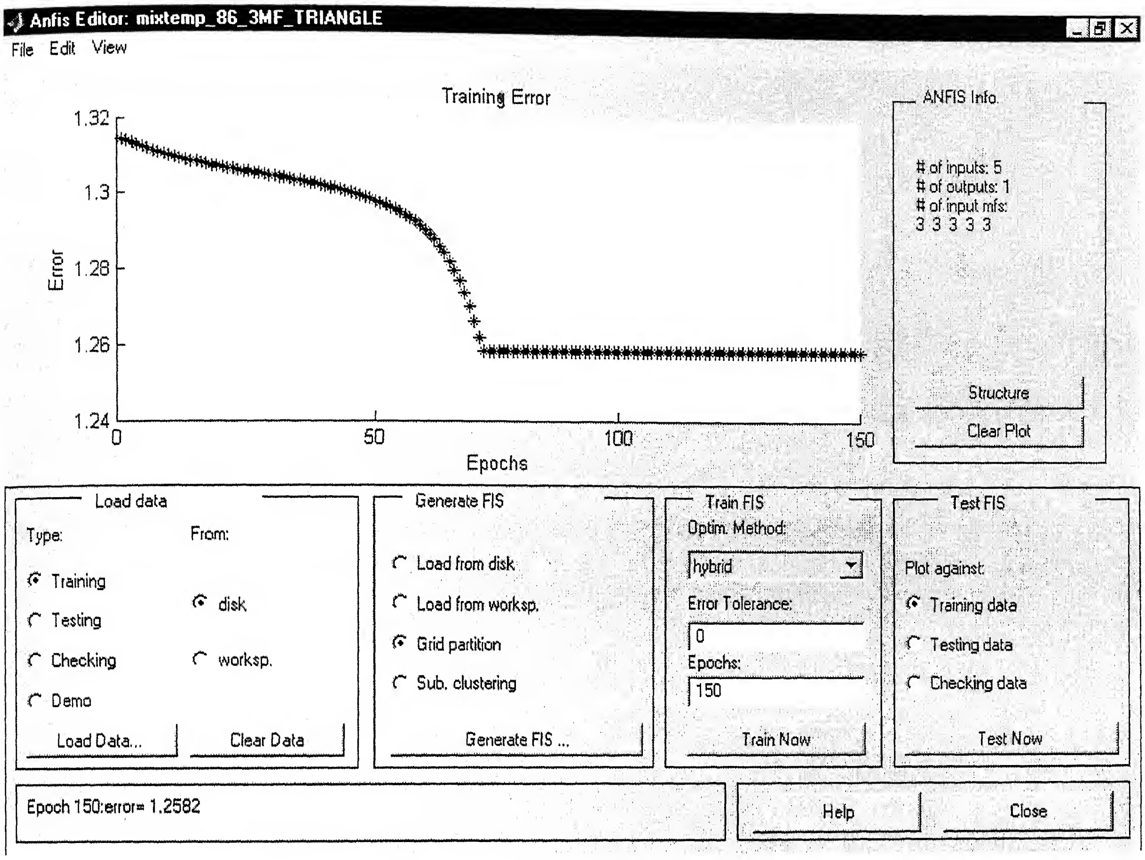


Fig 8.5 Training Completed and Training Error Saturates At 1.2582

➢ Once the Fis have been generated save them to the disk, as has been done in the case of RSP data of 86 and 117 data sets. A total of 8 Fis have been developed as per table 8.3 below.

8.2.6 Viewing/Editing the Mix Temperature Fis System Generated By ANFIS

Once the Fis system have been saved to the disk type fuzzy on the command window to open the fuzzy editor, and from the file menu open the Mix_Temp Fis and you will see the structure as in fig 8.6. In this one has to rename the input and output parameters as desired as by default they are named as input1, input2 and so on, in our case they have been renamed as CTI, GHDT, RT, MIXERTEMP, MIXINGTIME and MIXTEMP. The procedure of naming the parameters has already been explained in chapter 7.

NAME OF FIS	NO OF DATA POINTS	TYPE OF MEMBERSHIP FUNCTION	INPUTS WITH 3 MF EACH	OUTPUT WITH 3 MF
MIX_TEMP	86/117	TRIANGULAR	CTI	MIX_ TEMP
			GHDT	
			RT	
			MIXER TEMP	
			MIXING TIME	
GBD	86/117	TRIANGULAR	MIX_ TEMP	GBD
			ST	
			ROLLS	
			PMT	
			FP	
MIX_TEMP	86/117	TRAPEZOIDAL	CTI	MIX_ TEMP
			GHDT	
			RT	
			MIXER TEMP	
			MIXING TIME	
GBD	86/117	TRAPEZOIDAL	MIX_ TEMP	GBD
			ST	
			ROLLS	
			PMT	
			FP	

Table 8.3 Total of 8 Fis Systems Developed For RSP

8.2.6.1 Viewing/Editing the Membership Functions Generated By ANFIS

In order to view/edit the membership functions one selects edit membership functions and the following window will pop up showing all the membership functions that were generated for the input and output variables. As

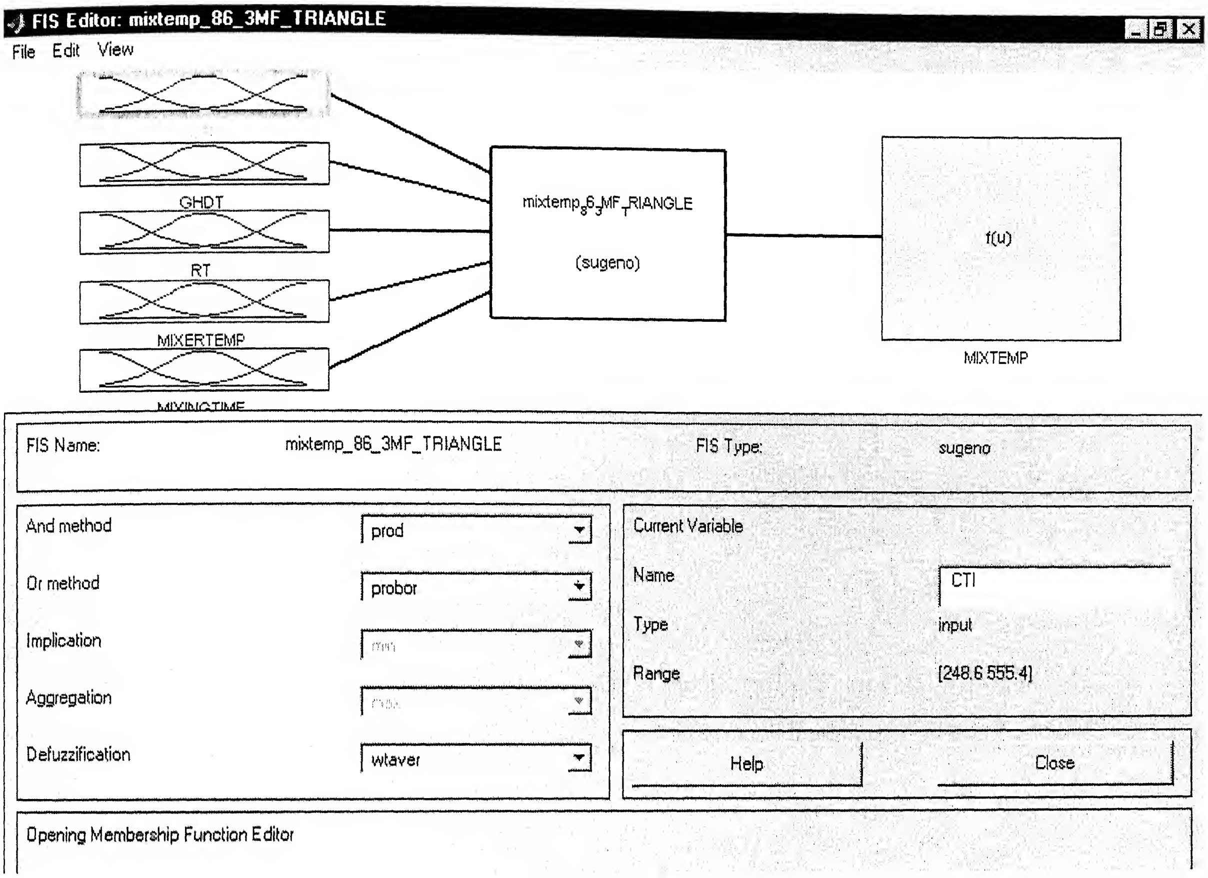


Fig 8.6 Fis System for Mix_Temp

a sample only the membership functions for CTI has been shown in fig 8.7 below. The membership functions for the remaining parameters GHDT, Retention Time, Mixer Temperature, Mixing time, are shown as figures 8.14 to 8.17 per Appendix K

8.2.6.2 Viewing/Editing the Rules Generated By ANFIS

For this one has to select edit rules and the following window is displayed as shown in fig 8.8 below. A total of $3^5 = 243$ rules are generated by the ANFIS editor. In case any of the rules are to be changed or modified they can be done as per procedure explained in chapter 7. In the case of RSP data model no rules have been modified as the results obtained were very encouraging. It may be noted that just by changing the variable names and the membership function names the rules get automatically updated to the new names.

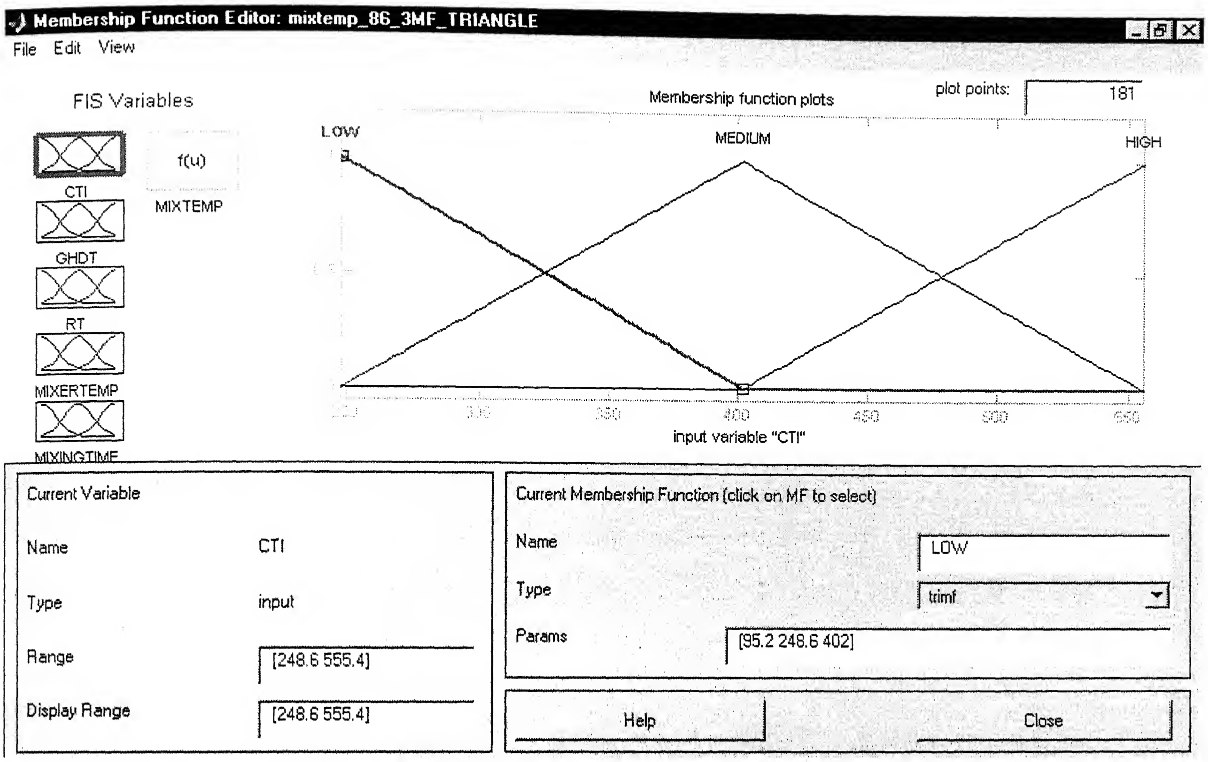


Fig 8.7 Membership functions for CTI generated by ANFIS editor

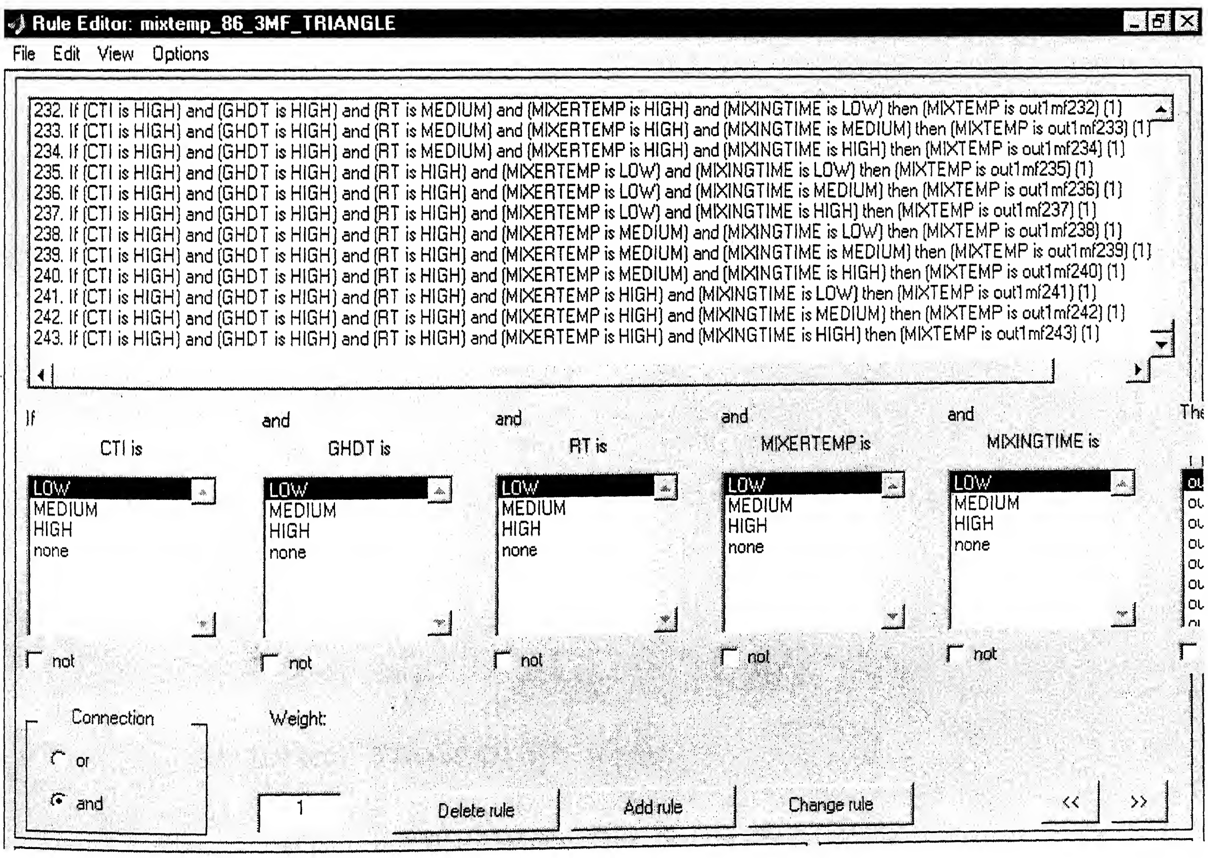


Fig 8.8 Rules Generated By ANFIS Editor Total Of $3^5 = 243$ Rules

8.2.6.3 Rule Viewer and Surface Viewer

The rule viewer and the surface viewer as already brought are read only windows and no editing can be done using them. But they give a bird's eye view of the relation between the inputs and out put. The windows generated on selecting these options are shown in figures 8.9 and 8.10. For the surface viewer only one of the combinations i.e. CTI and GHDT plotted against Mix temperature has been shown in the chapter the remaining have been attached as figures 8.18 to 8.20 of Appendix L.

8.2.7 Viewing/Editing the GBD Fis System Generated By ANFIS

On similar lines the Fis system for GBD has be generated by following the above steps and the results and out puts obtained have been attached as figures 8.21 to 8.32 in Appendix M.

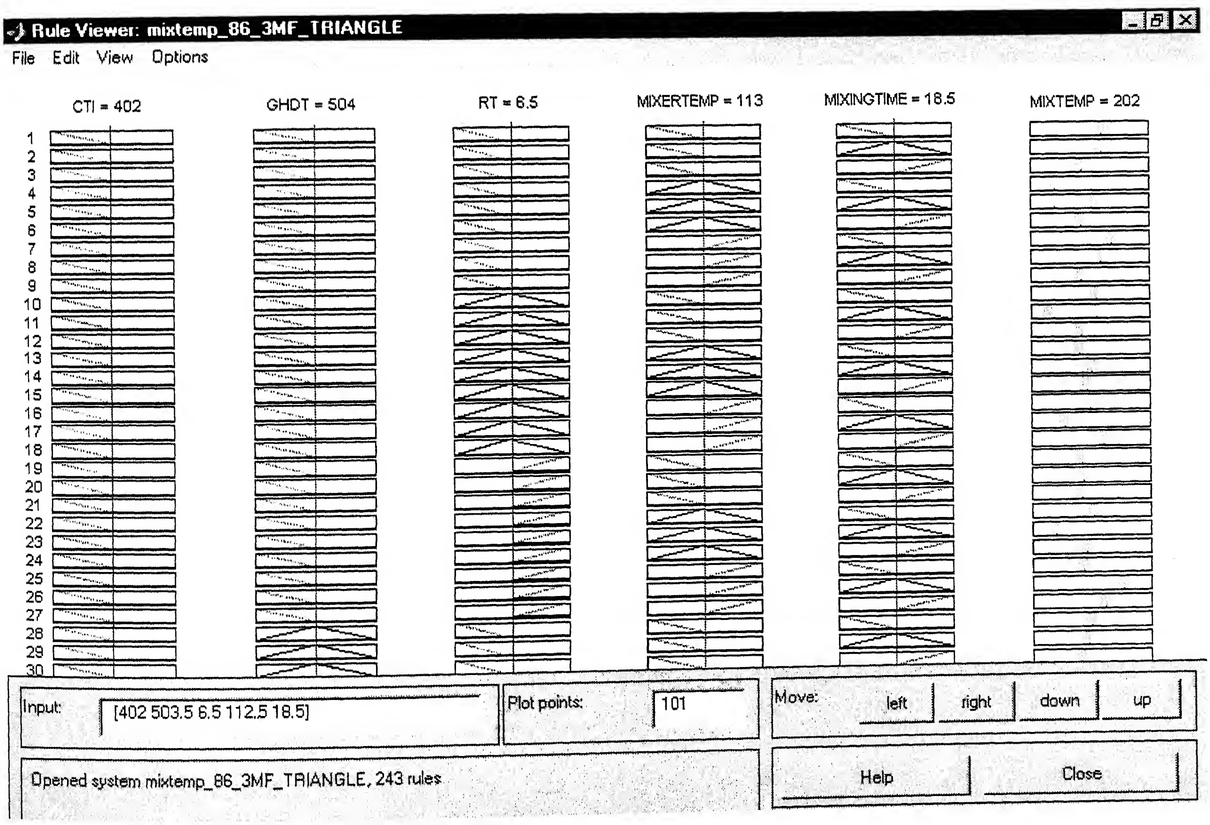


Fig 8.9 Rule Viewer for Mix temperature

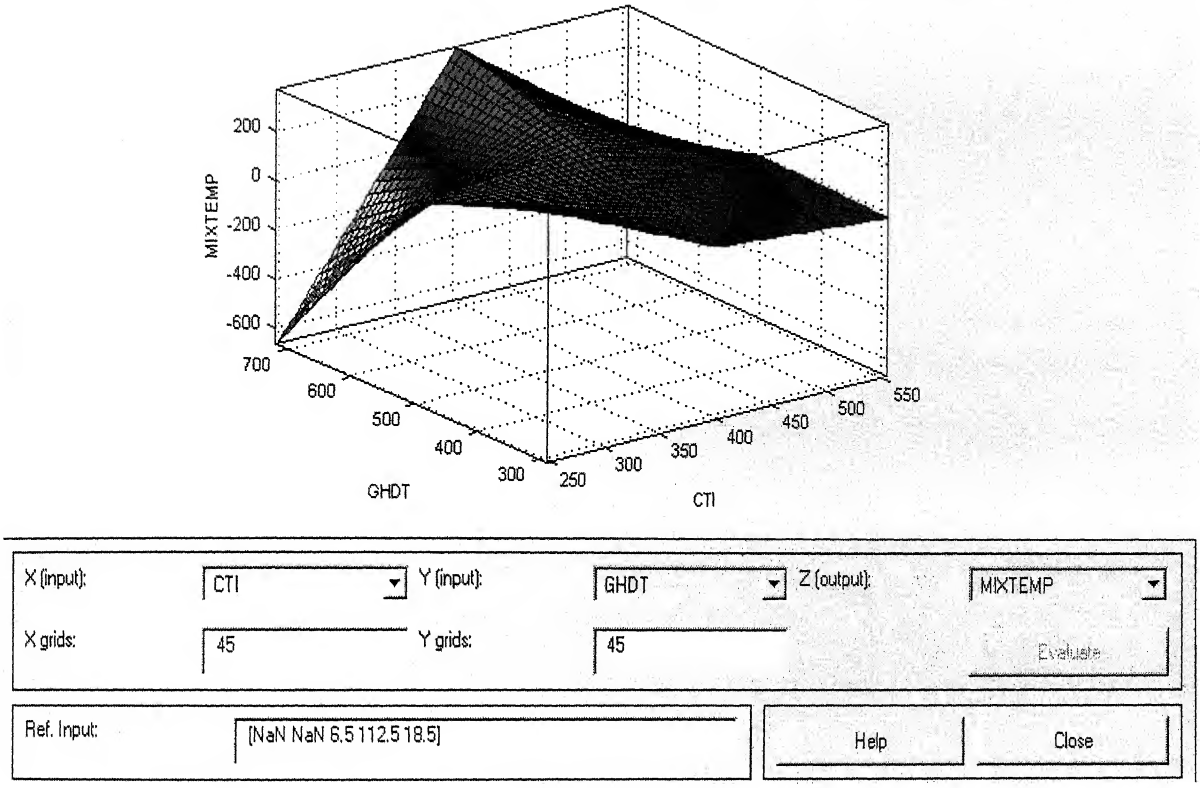


Fig 8.10 Surface Viewer for CTI and GHDT Plotted Against Mix Temperature

8.3 Validation Of Training Data Against Testing Data For 86 Data Set Vs 31 Data Set.

Once the training and testing data sets, Fis have been loaded from the disk now the test Fis radio buttons can be used to test the system for which Plot against testing data radio button is selected. The result obtained is shown in the figure 8.11 below. As can be seen from the plot which shows the desired out put in blue dots and the actual out put obtained from the Fis system trained by 86 data set with red crosses, that the out put obtained using the testing data has far to high an average error of **78.1146**, which confirms our contention that with just 86 data sets the system will not be able to capture the entire feature space of the process for which we require at least 3000 to 4000 data sets.

8.4 Results Obtained Using 117 Data Sets for Mix temperature Fis

The Fis systems generated using the 117 data set was tested and checked with the 86 and 31 data sets respectively. For this the test Fis was plotted against testing data and checking data and the results are shown in figure 8.12 and 8.13 respectively. It can be seen that the average error obtained has drastically dropped to 2.0939 and 2.9476 respectively. Which further substantiates the importance of training with larger volume of data?

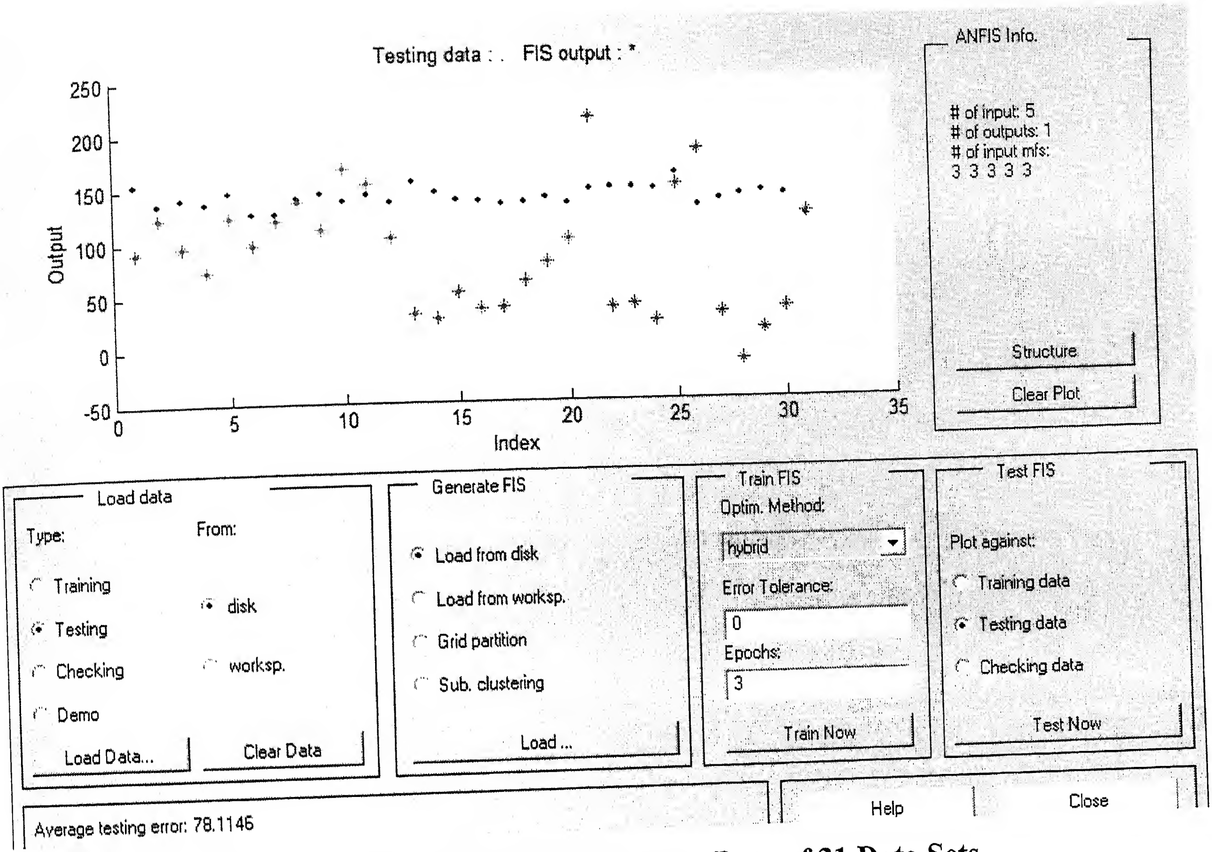


Fig 8.11 Testing the Fis System against Testing Data of 31 Data Sets

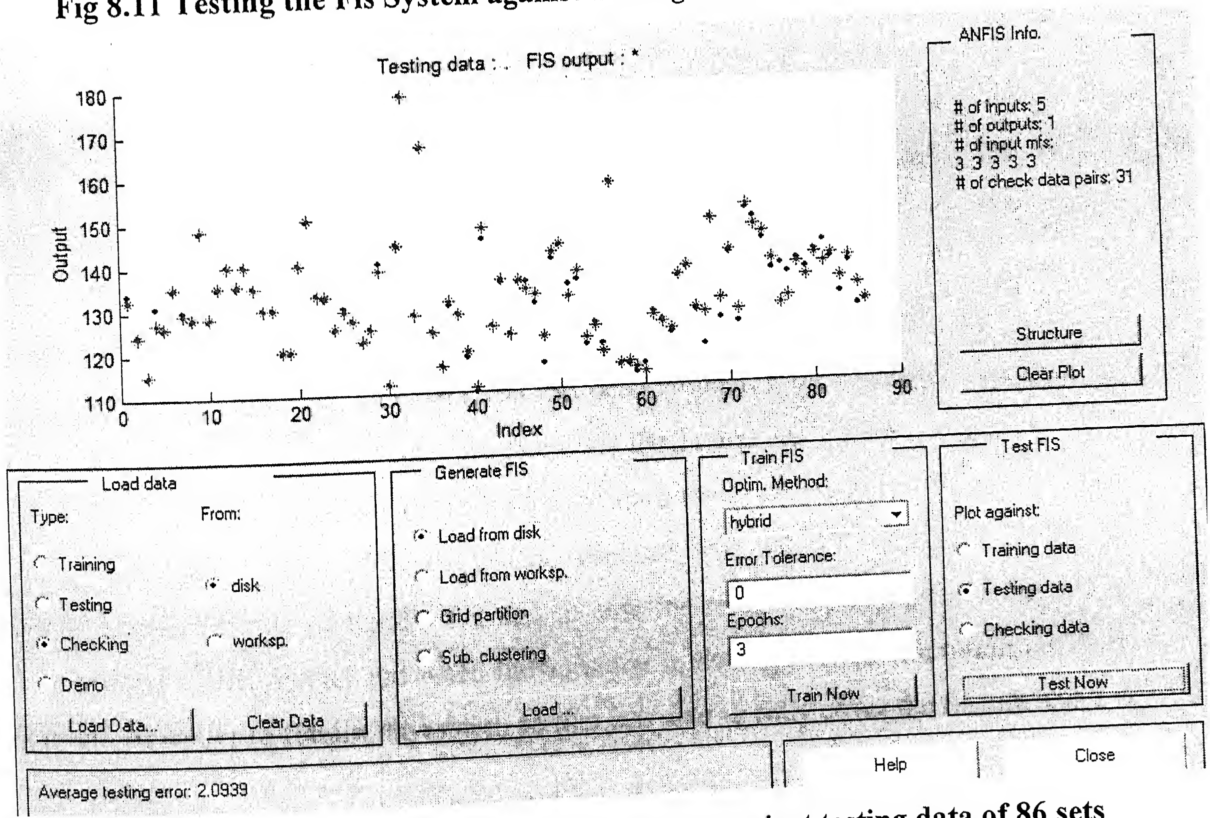


Fig 8.12 Results using Fis trained with117 data sets Plot against testing data of 86 sets

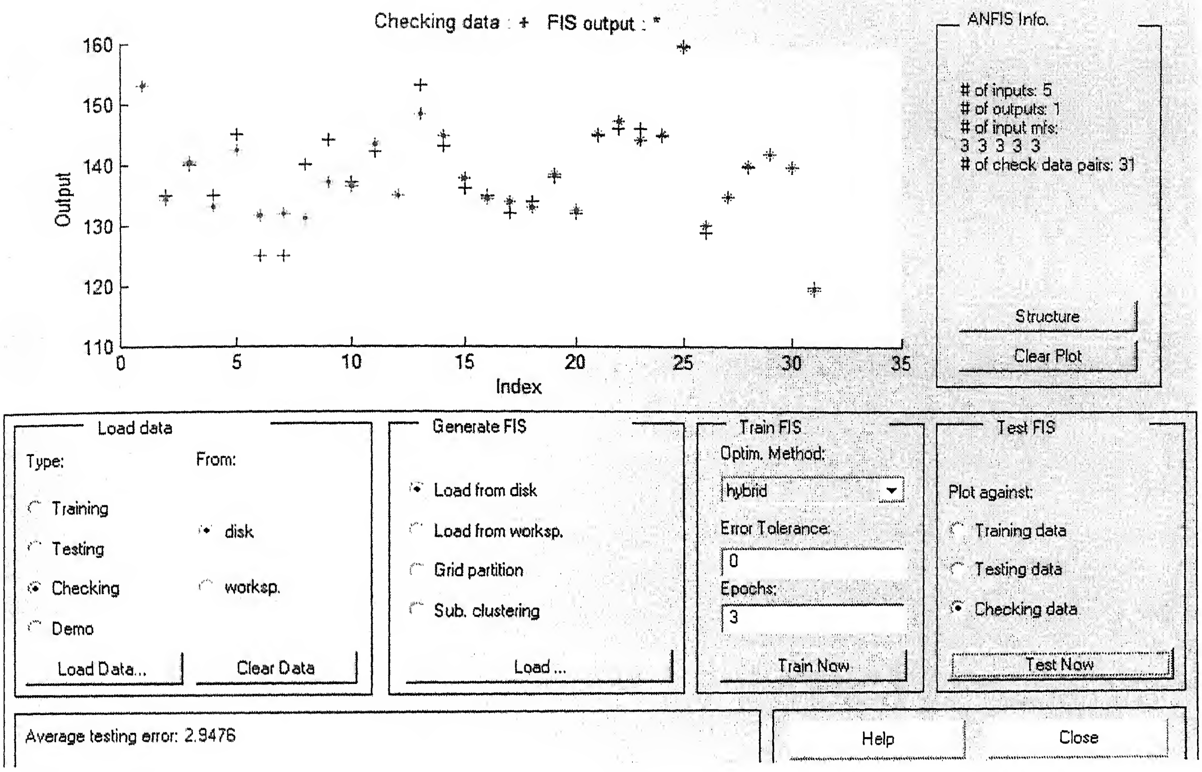


Fig 8.13 Results using Fis trained with117 data sets Plot against checking data of 31 sets

8.5 The Final Results for the Combined Systems

The Fis generated for Mix temperature and GBD were combined as explained earlier in this chapter using a matlab programme for the same. The programmes were also made to accept input for the user in vectors from or as a complete file from the keyboard itself. The outputs obtained for 86 data set Fis and That for 117 data set Fis are shown in table 8.4 and 8.5 attached as Appendix J. The 86 data set system results for Triangular membership and Trapezoidal membership have been compared with the Expert system developed in prolog as well. The out put of the systems developed for 117 data sets has been compared for Triangular membership and Trapezoidal membership. It has been observed that the Triangular system has more accurate and stable out put compared to that of trapezoidal system in both cases. When ever further new data sets get collected they should be test and checked with the 117 data set system for Validation as explained vide paragraph 8.3 and in case the results are satisfactory the system need not be trained afresh, but as and when the variation in data collected with minimum noise increases it will be better to retrain the system so that it is able to fully cover the feature space of the data distribution.

CHAPTER 9

COMPARISON BETWEEN THE MATLAB AND PROLOG BASED FUZZY INFERENCE SYSTEMS

9.1 Prolog based system

The prolog based FIS system is not equipped with the GUI and hence is not very user friendly for the plant operators [10]. It requires greater training for the operator to use the system effectively.

The prolog based system is not equipped with the surface view and rule viewer graphics that give an intuitive feel of the behavior of the input parameters and the rules that fire the system. This has its advantages as well in that the prolog system has a faster response to the input data as compared to the inputs to the Matlab system; this can have an effect in the future as the number of data collected increases many fold with the passage of time.

The prolog based system has only three types of membership functions, including the user defined function, compared to the matlab based system which can cater for eleven types of membership functions, excluding the user defined membership functions, as already brought out in chapter 7. However in the application for RSP only two types of the memberships could be used.

The implication method for prolog is only Mamdani, where as matlab has both Mamdani and Sugeno type and in addition user defined implications can also be used.

The Anfis editor of matlab is unique in that it generates the membership functions and rules on its own to fit the data input and has a built in safe guard to avoid over fitting. The only condition is that the data should be of low noise and of sufficient quantity to be able to be broken up into, separate training, teat and checking data. No such provision exists in the prolog system. All rule and membership functions have to be generated by the user/programmer himself.

The coding of programmes for matlab is very simple and user friendly and any changes to the programme can be implemented with ease and minimum amount of fuss. More over debugging of the programmes are very simple with extensive literature and

online help readily available for any needed assistance/help. The same can not be said of the prolog software.

The results obtained on test data has shown that the performance of both systems is satisfactory and hence result wise both are equally good.

9.2 Future Recommendation's

The Validation of both the systems developed has been held up for want of low noise data in sufficient quantity. This data should be made available as early as possible so as to fine tune the system either by modifying the membership functions or the rules base or both as the case may be. This may not be a very major job but is a very important one as only then can the system be put to any effective use.

The backward chaining systems developed in the Bayesian and FIS engines used for fault diagnoses have also to fine tuned for want of proper rules sets as they have not yet been provided by the experts nor is any tangible data available on the matter to generate these rules using tools such as Cart, ID3 etc.

The brick manufacturing process at RSP has a provision for it to be run on automation, which has not been done so far. The FIS system developed should be integrated to the automated system whenever it is made functional, only the will it be really possible to achieve the final target Brick life that the factory has in mind.

A graphic user interface should be developed for both the matlab and prolog systems so that the operator is able to run the desired modules in the programme by just a click of a button from his desktop.

9.3 Conclusion

In the mean while the FIS system developed can be used to train the operators on its use, so as to understand its capability and limitations. The much needed data to fine tune the system should be collected as early as possible for further processing. The control of the input parameters can still be achieved in the limited sense that the system has been able to learn form the however limited data that has already been used to develop and train the FIS system. The Expert based FIS can be fully implemented at this stage to monitor its performance even beyond the GDB stage.

EXPERT DASED DATA SET FOR RULE GENERATION

CTI	GHDT	RT	GT
HIGH	HIGH	HIGH	VERY HIGH
HIGH	HIGH	MEDIUM	HIGH
HIGH	HIGH	LOW	MEDIUM
HIGH	MEDIUM	HIGH	HIGH
HIGH	MEDIUM	MEDIUM	MEDIUM
HIGH	MEDIUM	LOW	MEDIUM
HIGH	LOW	HIGH	MEDIUM
HIGH	LOW	MEDIUM	LOW
HIGH	LOW	LOW	VERY LOW
MEDIUM	HIGH	HIGH	VERY HIGH
MEDIUM	HIGH	MEDIUM	HIGH
MEDIUM	HIGH	LOW	MEDIUM
MEDIUM	MEDIUM	HIGH	HIGH
MEDIUM	MEDIUM	MEDIUM	MEDIUM
MEDIUM	MEDIUM	LOW	LOW
MEDIUM	LOW	HIGH	MEDIUM
MEDIUM	LOW	MEDIUM	LOW
MEDIUM	LOW	LOW	VERY LOW
LOW	HIGH	HIGH	HIGH
LOW	HIGH	MEDIUM	MEDIUM
LOW	HIGH	LOW	LOW
LOW	MEDIUM	HIGH	MEDIUM
LOW	MEDIUM	MEDIUM	MEDIUM
LOW	MEDIUM	LOW	LOW
LOW	LOW	HIGH	LOW
LOW	LOW	MEDIUM	VERY LOW
LOW	LOW	LOW	VERY LOW

CTI
HIGH > 400
MEDIUM 300-400
LOW < 300

GHDT
HIGH > 500
MEDIUM 300-500
LOW < 300

RT
HIGH > 6
MEDIUM 4-6
LOW < 4

Table 4.2 for Grain Temperature & the ranges for High, Medium & Low

GT	MIXER TEMP	MIX TIME	MIX TEMP
VERY HIGH	HIGH	HIGH	**VERY HIGH**
VERY HIGH	HIGH	MEDIUM	**VERY HIGH**
VERY HIGH	HIGH	LOW	**VERY HIGH**
VERY HIGH	MEDIUM	HIGH	**VERY HIGH**
VERY HIGH	MEDIUM	MEDIUM	**VERY HIGH**
VERY HIGH	MEDIUM	LOW	**VERY HIGH**
VERY HIGH	LOW	HIGH	**VERY HIGH**
VERY HIGH	LOW	MEDIUM	**VERY HIGH**
VERY HIGH	LOW	LOW	**VERY HIGH**
HIGH	HIGH	HIGH	**HIGH**
HIGH	HIGH	MEDIUM	**HIGH**
HIGH	HIGH	LOW	**HIGH**
HIGH	MEDIUM	HIGH	**HIGH**
HIGH	MEDIUM	MEDIUM	**HIGH**
HIGH	MEDIUM	LOW	**HIGH**
HIGH	LOW	HIGH	**HIGH**
HIGH	LOW	MEDIUM	**HIGH**
HIGH	LOW	LOW	**MEDIUM**
MEDIUM	HIGH	HIGH	**MEDIUM**
MEDIUM	HIGH	MEDIUM	**MEDIUM**
MEDIUM	HIGH	LOW	**MEDIUM**
MEDIUM	MEDIUM	HIGH	**MEDIUM**
MEDIUM	MEDIUM	MEDIUM	**MEDIUM**
MEDIUM	MEDIUM	LOW	**MEDIUM**
MEDIUM	LOW	HIGH	**LOW**
MEDIUM	LOW	MEDIUM	**LOW**
MEDIUM	LOW	LOW	**MEDIUM**
LOW	HIGH	HIGH	**MEDIUM**
LOW	HIGH	MEDIUM	**LOW**
LOW	HIGH	LOW	**LOW**
LOW	MEDIUM	HIGH	**LOW**
LOW	MEDIUM	MEDIUM	**LOW**
LOW	MEDIUM	LOW	**LOW**
LOW	LOW	HIGH	**LOW**
LOW	LOW	MEDIUM	**LOW**
LOW	LOW	LOW	**LOW**
VERY LOW	HIGH	HIGH	**LOW**
VERY LOW	HIGH	MEDIUM	**LOW**
VERY LOW	HIGH	LOW	**VERY LOW**
VERY LOW	MEDIUM	HIGH	**LOW**
VERY LOW	MEDIUM	MEDIUM	**LOW**
VERY LOW	MEDIUM	LOW	**VERY LOW**
VERY LOW	LOW	HIGH	**VERY LOW**
VERY LOW	LOW	MEDIUM	**VERY LOW**
VERY LOW	LOW	LOW	**VERY LOW**

MIXER TEMP
HIGH > 120
MEDIUM 100-120
LOW < 100

MIXING TIME
HIGH > 20
MEDIUM 10-20
LOW < 10

Table 4.3 for Mix Temperature & the ranges for High, Medium & Low

MIX TEMP	ST	ROLLINGS	PRESS MIX TEMP
VERY HIGH	HIGH	HIGH	LOW
VERY HIGH	HIGH	MEDIUM	MEDIUM
VERY HIGH	HIGH	LOW	HIGH
VERY HIGH	MEDIUM	HIGH	LOW
VERY HIGH	MEDIUM	MEDIUM	MEDIUM
VERY HIGH	MEDIUM	LOW	HIGH
VERY HIGH	LOW	HIGH	MEDIUM
VERY HIGH	LOW	MEDIUM	HIGH
VERY HIGH	LOW	LOW	VERY HIGH
HIGH	HIGH	HIGH	MEDIUM
HIGH	HIGH	MEDIUM	MEDIUM
HIGH	HIGH	LOW	HIGH
HIGH	MEDIUM	HIGH	MEDIUM
HIGH	MEDIUM	MEDIUM	MEDIUM
HIGH	MEDIUM	LOW	HIGH
HIGH	LOW	HIGH	MEDIUM
HIGH	LOW	MEDIUM	HIGH
HIGH	LOW	LOW	HIGH
MEDIUM	HIGH	HIGH	LOW
MEDIUM	HIGH	MEDIUM	LOW
MEDIUM	HIGH	LOW	MEDIUM
MEDIUM	MEDIUM	HIGH	LOW
MEDIUM	MEDIUM	MEDIUM	LOW
MEDIUM	MEDIUM	LOW	MEDIUM
MEDIUM	LOW	HIGH	LOW
MEDIUM	LOW	MEDIUM	LOW
MEDIUM	LOW	LOW	MEDIUM
LOW	HIGH	HIGH	LOW
LOW	HIGH	MEDIUM	LOW
LOW	HIGH	LOW	LOW
LOW	MEDIUM	HIGH	LOW
LOW	MEDIUM	MEDIUM	LOW
LOW	MEDIUM	LOW	LOW
LOW	LOW	HIGH	LOW
LOW	LOW	MEDIUM	LOW
LOW	LOW	LOW	LOW
VERY LOW	HIGH	HIGH	VERY LOW
VERY LOW	HIGH	MEDIUM	VERY LOW
VERY LOW	HIGH	LOW	VERY LOW
VERY LOW	MEDIUM	HIGH	VERY LOW
VERY LOW	MEDIUM	MEDIUM	VERY LOW
VERY LOW	MEDIUM	LOW	LOW
VERY LOW	LOW	HIGH	VERY LOW
VERY LOW	LOW	MEDIUM	VERY LOW
VERY LOW	LOW	LOW	LOW

STORAGE TIME
HIGH > 90
MEDIUM 30-90
LOW < 30

NO. OF ROLLINGS
HIGH > 2
MEDIUM 1-2
LOW < 1

TABLE4.4 for PMT & the ranges for High, Medium & Low

FP	PMT	PITCH	GRAPHITE	GREEN BD
HIGH	HIGH	HIGH	VERY HIGH	LOW
HIGH	HIGH	HIGH	HIGH	MEDIUM
HIGH	HIGH	HIGH	MEDIUM	MEDIUM
HIGH	HIGH	HIGH	LOW	HIGH
HIGH	HIGH	HIGH	VERY LOW	VERY HIGH
HIGH	HIGH	MEDIUM	VERY HIGH	LOW
HIGH	HIGH	MEDIUM	HIGH	LOW
HIGH	HIGH	MEDIUM	MEDIUM	MEDIUM
HIGH	HIGH	MEDIUM	LOW	HIGH
HIGH	HIGH	MEDIUM	VERY LOW	VERY HIGH
HIGH	HIGH	LOW	VERY HIGH	VERY LOW
HIGH	HIGH	LOW	HIGH	LOW
HIGH	HIGH	LOW	MEDIUM	MEDIUM
HIGH	HIGH	LOW	LOW	MEDIUM
HIGH	HIGH	LOW	VERY LOW	HIGH
HIGH	MEDIUM	HIGH	VERY HIGH	LOW
HIGH	MEDIUM	HIGH	HIGH	LOW
HIGH	MEDIUM	HIGH	MEDIUM	MEDIUM
HIGH	MEDIUM	HIGH	LOW	HIGH
HIGH	MEDIUM	HIGH	VERY LOW	HIGH
HIGH	MEDIUM	MEDIUM	VERY HIGH	VERY LOW
HIGH	MEDIUM	MEDIUM	HIGH	LOW
HIGH	MEDIUM	MEDIUM	MEDIUM	MEDIUM
HIGH	MEDIUM	MEDIUM	LOW	MEDIUM
HIGH	MEDIUM	MEDIUM	VERY LOW	HIGH
HIGH	MEDIUM	LOW	VERY HIGH	VERY LOW
HIGH	MEDIUM	LOW	HIGH	LOW
HIGH	MEDIUM	LOW	MEDIUM	MEDIUM
HIGH	MEDIUM	LOW	LOW	MEDIUM
HIGH	MEDIUM	LOW	VERY LOW	HIGH
HIGH	LOW	HIGH	VERY HIGH	VERY LOW
HIGH	LOW	HIGH	HIGH	LOW
HIGH	LOW	HIGH	MEDIUM	LOW
HIGH	LOW	HIGH	LOW	MEDIUM
HIGH	LOW	HIGH	VERY LOW	MEDIUM
HIGH	LOW	MEDIUM	VERY HIGH	VERY LOW
HIGH	LOW	MEDIUM	HIGH	LOW
HIGH	LOW	MEDIUM	MEDIUM	MEDIUM
HIGH	LOW	MEDIUM	LOW	MEDIUM
HIGH	LOW	MEDIUM	VERY LOW	HIGH
HIGH	LOW	LOW	VERY HIGH	VERY LOW
HIGH	LOW	LOW	HIGH	LOW
HIGH	LOW	LOW	MEDIUM	LOW
HIGH	LOW	LOW	LOW	MEDIUM
HIGH	LOW	LOW	LOW	MEDIUM

PITCH
HIGH > 4.8
MEDIUM 4.2-4.8
LOW < 4.2

GRAPHITE
VERY HIGH > 7
HIGH = 5.6-7
MEDIUM=4.5-5.5
LOW = 3.0-4.4
VERY LOW < 3

PITCH
HIGH > 4.8
MEDIUM 4.2-4.8
LOW < 4.2

GRAPHITE
VERY HIGH > 7
HIGH = 5.6-7
MEDIUM=4.5-5.5
LOW = 3.0-4.4
VERY LOW < 3

TABLE4.5Table for GBD & the ranges for High, Medium & Low(1of 2)

FP	PMT	PITCH	GRAPHITE	GREEN BD
LOW	HIGH	HIGH	VERY HIGH	VERY LOW
LOW	HIGH	HIGH	HIGH	LOW
LOW	HIGH	HIGH	MEDIUM	MEDIUM
LOW	HIGH	HIGH	LOW	MEDIUM
LOW	HIGH	HIGH	VERY LOW	HIGH
LOW	HIGH	MEDIUM	VERY HIGH	VERY LOW
LOW	HIGH	MEDIUM	HIGH	VERY LOW
LOW	HIGH	MEDIUM	MEDIUM	LOW
LOW	HIGH	MEDIUM	LOW	MEDIUM
LOW	HIGH	MEDIUM	VERY LOW	HIGH
LOW	HIGH	LOW	VERY HIGH	VERY LOW
LOW	HIGH	LOW	HIGH	VERY LOW
LOW	HIGH	LOW	MEDIUM	LOW
LOW	HIGH	LOW	LOW	LOW
LOW	HIGH	LOW	VERY LOW	MEDIUM
LOW	MEDIUM	HIGH	VERY HIGH	VERY LOW
LOW	MEDIUM	HIGH	HIGH	LOW
LOW	MEDIUM	HIGH	MEDIUM	MEDIUM
LOW	MEDIUM	HIGH	LOW	HIGH
LOW	MEDIUM	HIGH	VERY LOW	HIGH
LOW	MEDIUM	MEDIUM	VERY HIGH	VERY LOW
LOW	MEDIUM	MEDIUM	HIGH	VERY LOW
LOW	MEDIUM	MEDIUM	MEDIUM	LOW
LOW	MEDIUM	MEDIUM	LOW	MEDIUM
LOW	MEDIUM	MEDIUM	VERY LOW	MEDIUM
LOW	MEDIUM	LOW	VERY HIGH	VERY LOW
LOW	MEDIUM	LOW	HIGH	VERY LOW
LOW	MEDIUM	LOW	MEDIUM	LOW
LOW	MEDIUM	LOW	LOW	MEDIUM
LOW	MEDIUM	LOW	VERY LOW	MEDIUM
LOW	LOW	HIGH	VERY HIGH	VERY LOW
LOW	LOW	HIGH	HIGH	VERY LOW
LOW	LOW	HIGH	MEDIUM	LOW
LOW	LOW	HIGH	LOW	LOW
LOW	LOW	HIGH	VERY LOW	MEDIUM
LOW	LOW	MEDIUM	VERY HIGH	VERY LOW
LOW	LOW	MEDIUM	HIGH	VERY LOW
LOW	LOW	MEDIUM	MEDIUM	LOW
LOW	LOW	MEDIUM	LOW	MEDIUM
LOW	LOW	MEDIUM	VERY LOW	MEDIUM
LOW	LOW	LOW	VERY HIGH	VERY LOW
LOW	LOW	LOW	HIGH	VERY LOW
LOW	LOW	LOW	MEDIUM	VERY LOW
LOW	LOW	LOW	LOW	LOW
LOW	LOW	LOW	LOW	LOW

TABLE4.5 for GBD & the ranges for High, Medium & Low(2of 2)

PITCH	TEMPERING	CCS
HIGH	PERFECT	VERY_HIGH
HIGH	UNDER	VERY_HIGH
HIGH	OVER	HIGH
MEDIUM	PERFECT	VERY_HIGH
MEDIUM	UNDER	HIGH
MEDIUM	OVER	HIGH
LOW	PERFECT	HIGH
LOW	UNDER	MEDIUM
LOW	OVER	VERY_HIGH
HIGH	PERFECT	VERY_HIGH
HIGH	UNDER	HIGH
HIGH	OVER	HIGH
MEDIUM	PERFECT	HIGH
MEDIUM	UNDER	MEDIUM
MEDIUM	OVER	MEDIUM
LOW	PERFECT	MEDIUM
LOW	UNDER	MEDIUM
LOW	OVER	MEDIUM
HIGH	PERFECT	HIGH
HIGH	UNDER	HIGH
HIGH	OVER	MEDIUM
MEDIUM	PERFECT	MEDIUM
MEDIUM	UNDER	MEDIUM
MEDIUM	OVER	LOW
LOW	PERFECT	LOW
LOW	UNDER	MEDIUM
LOW	OVER	LOW
HIGH	PERFECT	MEDIUM
HIGH	UNDER	MEDIUM
HIGH	OVER	LOW
MEDIUM	PERFECT	MEDIUM
MEDIUM	UNDER	MEDIUM
MEDIUM	OVER	LOW
LOW	PERFECT	LOW
LOW	UNDER	LOW
LOW	OVER	VERY_LOW
HIGH	PERFECT	VERY_LOW

TABLE4.6 for Rules of CCS(1 of 2)

PITCH	TEMPERING	CCS
HIGH	UNDER	LOW
HIGH	OVER	VERY_LOW
MEDIUM	PERFECT	VERY_LOW
MEDIUM	UNDER	LOW
MEDIUM	OVER	VERY_LOW
LOW	PERFECT	VERY_LOW
LOW	UNDER	VERY_LOW
LOW	OVER	VERY_LOW

TABLE4.6 for Rules of CCS(2 of 2)

GREEN BD	PITCH QUANTITY	COKED POROSITY
VERY_HIGH	HIGH	MEDIUM
VERY_HIGH	MEDIUM	LOW
VERY_HIGH	LOW	LOW
HIGH	HIGH	HIGH
HIGH	MEDIUM	MEDIUM
HIGH	LOW	LOW
MEDIUM	HIGH	HIGH
MEDIUM	MEDIUM	MEDIUM
MEDIUM	LOW	MEDIUM
LOW	HIGH	HIGH
LOW	MEDIUM	MEDIUM
LOW	LOW	MEDIUM
VERY_LOW	HIGH	HIGH
VERY_LOW	MEDIUM	HIGH
VERY_LOW	LOW	HIGH

TABLE4.7 for Rules of CP

CARBON CONTENT	GREEN BD	METAL POWDER	COLD CRUSHING STRENGTH	HMOR
VERY_HIGH	LOW	HIGH	HIGH	VERY_VERY_HIGH
VERY_HIGH	LOW	HIGH	MEDIUM	VERY_VERY_HIGH
VERY_HIGH	LOW	HIGH	LOW	VERY_VERY_HIGH
VERY_HIGH	LOW	MEDIUM	HIGH	VERY_VERY_HIGH
VERY_HIGH	LOW	MEDIUM	MEDIUM	VERY_VERY_HIGH
VERY_HIGH	LOW	MEDIUM	LOW	VERY_HIGH
VERY_HIGH	LOW	LOW	HIGH	HIGH
VERY_HIGH	LOW	LOW	MEDIUM	HIGH
VERY_HIGH	LOW	LOW	LOW	MEDIUM
VERY_HIGH	VERY_LOW	HIGH	HIGH	VERY_VERY_HIGH
VERY_HIGH	VERY_LOW	HIGH	MEDIUM	VERY_HIGH
VERY_HIGH	VERY_LOW	HIGH	LOW	VERY_HIGH
VERY_HIGH	VERY_LOW	MEDIUM	HIGH	HIGH
VERY_HIGH	VERY_LOW	MEDIUM	MEDIUM	HIGH
VERY_HIGH	VERY_LOW	MEDIUM	LOW	HIGH
VERY_HIGH	VERY_LOW	LOW	HIGH	MEDIUM
VERY_HIGH	VERY_LOW	LOW	MEDIUM	MEDIUM
VERY_HIGH	VERY_LOW	LOW	LOW	MEDIUM
HIGH	MEDIUM	HIGH	HIGH	VERY_VERY_HIGH
HIGH	MEDIUM	HIGH	MEDIUM	VERY_VERY_HIGH
HIGH	MEDIUM	HIGH	LOW	VERY_VERY_HIGH
HIGH	MEDIUM	MEDIUM	HIGH	VERY_VERY_HIGH
HIGH	MEDIUM	MEDIUM	MEDIUM	VERY_VERY_HIGH
HIGH	MEDIUM	MEDIUM	LOW	VERY_HIGH
HIGH	MEDIUM	LOW	HIGH	HIGH
HIGH	MEDIUM	LOW	MEDIUM	MEDIUM
HIGH	MEDIUM	LOW	LOW	MEDIUM
HIGH	LOW	HIGH	HIGH	VERY_VERY_HIGH
HIGH	LOW	HIGH	MEDIUM	VERY_VERY_HIGH
HIGH	LOW	HIGH	LOW	VERY_HIGH
HIGH	LOW	MEDIUM	HIGH	VERY_HIGH
HIGH	LOW	MEDIUM	MEDIUM	VERY_HIGH
HIGH	LOW	MEDIUM	LOW	HIGH
HIGH	LOW	LOW	HIGH	HIGH

TABLE4.8 RULES FOR HMOR(1 of 3)

HIGH	LOW	LOW	MEDIUM	MEDIUM
HIGH	LOW	LOW	LOW	MEDIUM
HIGH	VERY_LOW	HIGH	HIGH	VERY_HIGH
HIGH	VERY_LOW	HIGH	MEDIUM	VERY_HIGH
HIGH	VERY_LOW	HIGH	LOW	VERY_HIGH
HIGH	VERY_LOW	MEDIUM	HIGH	HIGH
HIGH	VERY_LOW	MEDIUM	MEDIUM	HIGH
HIGH	VERY_LOW	MEDIUM	LOW	MEDIUM
HIGH	VERY_LOW	LOW	HIGH	MEDIUM
HIGH	VERY_LOW	LOW	MEDIUM	MEDIUM
HIGH	VERY_LOW	LOW	LOW	MEDIUM
MEDIUM	MEDIUM	HIGH	HIGH	VERY_VERY_HIGH
MEDIUM	MEDIUM	HIGH	MEDIUM	VERY_VERY_HIGH
MEDIUM	MEDIUM	HIGH	LOW	VERY_VERY_HIGH
MEDIUM	MEDIUM	MEDIUM	HIGH	VERY_VERY_HIGH
MEDIUM	MEDIUM	MEDIUM	MEDIUM	VERY_HIGH
MEDIUM	MEDIUM	MEDIUM	LOW	HIGH
MEDIUM	MEDIUM	LOW	HIGH	MEDIUM
MEDIUM	MEDIUM	LOW	MEDIUM	MEDIUM
MEDIUM	MEDIUM	LOW	LOW	LOW
MEDIUM	LOW	HIGH	HIGH	VERY_HIGH
MEDIUM	LOW	HIGH	MEDIUM	VERY_HIGH
MEDIUM	LOW	HIGH	LOW	VERY_HIGH
MEDIUM	LOW	MEDIUM	HIGH	VERY_HIGH
MEDIUM	LOW	MEDIUM	MEDIUM	VERY_HIGH
MEDIUM	LOW	MEDIUM	LOW	HIGH
MEDIUM	LOW	LOW	HIGH	MEDIUM
MEDIUM	LOW	LOW	MEDIUM	MEDIUM
MEDIUM	LOW	LOW	LOW	LOW
MEDIUM	VERY_LOW	HIGH	HIGH	HIGH
MEDIUM	VERY_LOW	HIGH	MEDIUM	MEDIUM
MEDIUM	VERY_LOW	HIGH	LOW	MEDIUM
MEDIUM	VERY_LOW	MEDIUM	HIGH	MEDIUM
MEDIUM	VERY_LOW	MEDIUM	MEDIUM	MEDIUM
MEDIUM	VERY_LOW	MEDIUM	LOW	LOW
MEDIUM	VERY_LOW	LOW	HIGH	LOW
MEDIUM	VERY_LOW	LOW	MEDIUM	VERY_LOW

TABLE4.8 RULES FOR HMOR(2of 3)

GREEN BD	CARBON CONTENT	METAL POWDER	OXIDATION RESISTANCE
VERY_HIGH	VERY_LOW	HIGH	MEDIUM
VERY_HIGH	VERY_LOW	MEDIUM	LOW
VERY_HIGH	VERY_LOW	LOW	VERY_LOW
HIGH	VERY_LOW	HIGH	LOW
HIGH	VERY_LOW	MEDIUM	VERY_LOW
HIGH	VERY_LOW	LOW	VERY_LOW
HIGH	LOW	HIGH	MEDIUM
HIGH	LOW	MEDIUM	LOW
HIGH	LOW	LOW	LOW
MEDIUM	HIGH	HIGH	VERY_HIGH
MEDIUM	HIGH	MEDIUM	HIGH
MEDIUM	HIGH	LOW	MEDIUM
MEDIUM	MEDIUM	HIGH	HIGH
MEDIUM	MEDIUM	MEDIUM	HIGH
MEDIUM	MEDIUM	LOW	MEDIUM
MEDIUM	LOW	HIGH	MEDIUM
MEDIUM	LOW	MEDIUM	LOW
MEDIUM	LOW	LOW	LOW
MEDIUM	VERY_LOW	HIGH	LOW
MEDIUM	VERY_LOW	MEDIUM	LOW
MEDIUM	VERY_LOW	LOW	VERY_LOW
LOW	VERY_HIGH	HIGH	VERY_HIGH
LOW	VERY_HIGH	MEDIUM	VERY_HIGH
LOW	VERY_HIGH	LOW	HIGH
LOW	VERY_LOW	HIGH	LOW
LOW	VERY_LOW	MEDIUM	VERY_LOW
LOW	VERY_LOW	LOW	VERY_LOW
LOW	HIGH	HIGH	HIGH
LOW	HIGH	MEDIUM	HIGH
LOW	HIGH	LOW	MEDIUM
LOW	MEDIUM	HIGH	MEDIUM
LOW	MEDIUM	MEDIUM	LOW

TABLE4.9 RULES FOR OXIDATION RESISTANCE(1 of 2)

LOW	MEDIUM	LOW	LOW
LOW	LOW	HIGH	LOW
LOW	LOW	MEDIUM	VERY_LOW
LOW	LOW	LOW	VERY_LOW
VERY_LOW	HIGH	HIGH	MEDIUM
VERY_LOW	HIGH	MEDIUM	MEDIUM
VERY_LOW	HIGH	LOW	LOW
VERY_LOW	MEDIUM	HIGH	MEDIUM
VERY_LOW	MEDIUM	MEDIUM	LOW
VERY_LOW	MEDIUM	LOW	LOW
VERY_LOW	VERY_HIGH	HIGH	VERY_HIGH
VERY_LOW	VERY_HIGH	MEDIUM	HIGH
VERY_LOW	VERY_HIGH	LOW	MEDIUM

TABLE4.9 RULES FOR OXIDATION RESISTANCE (2 of 2)

CARBON CONTENT	CHEMICAL PURITY	COKED POROSITY	SLAG CORROSION RESISTANCE
MEDIUM	HIGH	HIGH	MEDIUM
MEDIUM	HIGH	MEDIUM	HIGH
MEDIUM	HIGH	LOW	HIGH
MEDIUM	MEDIUM	HIGH	MEDIUM
MEDIUM	MEDIUM	MEDIUM	MEDIUM
MEDIUM	MEDIUM	LOW	HIGH
MEDIUM	LOW	HIGH	LOW
MEDIUM	LOW	MEDIUM	LOW
MEDIUM	LOW	LOW	LOW
LOW	HIGH	HIGH	LOW
LOW	HIGH	MEDIUM	MEDIUM
LOW	HIGH	LOW	MEDIUM
LOW	MEDIUM	HIGH	LOW
LOW	MEDIUM	MEDIUM	MEDIUM
LOW	MEDIUM	LOW	MEDIUM
LOW	LOW	HIGH	VERY_LOW
LOW	LOW	MEDIUM	VERY_LOW
LOW	LOW	LOW	LOW

TABLE4.10 RULES FOR SLAG CORROSION RESISTANCE(1of 2)

CARBON CONTENT	CHEMICAL PURITY	COKED POROSITY	SLAG CORROSION RESISTANCE
VERY LOW	HIGH	HIGH	LOW
VERY LOW	HIGH	MEDIUM	MEDIUM
VERY LOW	HIGH	LOW	MEDIUM
VERY LOW	MEDIUM	HIGH	VERY_LOW
VERY LOW	MEDIUM	MEDIUM	LOW
VERY LOW	MEDIUM	LOW	LOW
VERY LOW	LOW	HIGH	VERY_LOW
VERY LOW	LOW	MEDIUM	VERY_LOW
VERY LOW	LOW	LOW	LOW
VERY HIGH	HIGH	HIGH	HIGH
VERY HIGH	HIGH	MEDIUM	VERY_HIGH
VERY HIGH	HIGH	LOW	VERY_HIGH
VERY HIGH	MEDIUM	HIGH	HIGH
VERY HIGH	MEDIUM	MEDIUM	HIGH
VERY HIGH	MEDIUM	LOW	VERY_HIGH
VERY HIGH	LOW	HIGH	LOW
VERY HIGH	LOW	MEDIUM	MEDIUM
VERY HIGH	LOW	LOW	MEDIUM
HIGH	HIGH	HIGH	HIGH
HIGH	HIGH	MEDIUM	HIGH
HIGH	HIGH	LOW	VERY_HIGH
HIGH	MEDIUM	HIGH	HIGH
HIGH	MEDIUM	MEDIUM	HIGH
HIGH	MEDIUM	LOW	VERY_HIGH
HIGH	LOW	HIGH	LOW
HIGH	LOW	MEDIUM	LOW
HIGH	LOW	LOW	MEDIUM

TABLE4.10 RULES FOR SLAG CORROSION RESISTANCE (2of 2)

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLLS	FP	GBD
low	high	low	high	medium	medium	high	medium	low	medium	medium
low	low	low	medium	medium	low	very_low	high	medium	medium	low
medium	medium	medium	low	high	low	very_low	medium	low	low	very_low
low	low	high	high	medium	medium	medium	medium	medium	high	high
low	medium	high	high	high	medium	medium	high	low	high	medium
high	high	low	high	high	medium	medium	medium	medium	high	medium
medium	high	low	high	medium	medium	medium	high	low	medium	low
high	low	low	high	high	medium	low	low	medium	low	medium
medium	low	high	high	high	high	medium	medium	medium	low	very_low

Table 4.11 Actual Data of 86 sets Obtained from RSP Converted to VeryHigh, High, Medium, low , verylow (1 of 3)

high	medium	medium	high	low	medium	high	medium	low	high	medium
high	high	medium	low	medium	medium	low	low	medium	medium	medium
high	high	low	low	medium	high	high	low	medium	high	medium
high	medium	low	high	medium	medium	medium	medium	medium	high	medium
high	medium	low	high	medium	high	high	medium	medium	high	medium
high	medium	low	medium	medium	medium	medium	medium	medium	high	low
low	low	low	low	medium	medium	medium	medium	low	high	high
low	low	medium	low	medium	medium	low	medium	medium	high	low
low	low	high	low	medium	low	medium	medium	low	high	medium
low	low	medium	low	medium	low	medium	medium	low	high	low
high	medium	low	medium	medium	high	high	high	low	medium	Very_low
high	low	low	high	low	high	high	high	medium	medium	low
high	medium	low	low	low	medium	high	medium	low	medium	low
high	medium	low	low	low	medium	high	medium	low	medium	medium
high	low	low	medium	medium	low	medium	medium	low	low	low
medium	low	medium	medium	medium	medium	medium	low	medium	medium	high
high	low	medium	high	low	medium	medium	medium	low	medium	medium
high	low	low	high	low	low	medium	low	low	medium	medium
high	low	low	medium	low	low	medium	medium	low	medium	medium
medium	low	high	medium	medium	high	medium	high	low	high	medium
medium	high	low	low	high	low	very_low	low	low	medium	medium
high	low	low	high	high	high	medium	high	low	medium	very_low
high	low	low	high	medium	high	high	high	medium	medium	very_low
low	low	high	low	medium	medium	high	low	low	medium	high
low	high	high	low	low	high	high	low	medium	medium	high
low	high	medium	medium	low	low	medium	medium	low	low	medium
low	medium	low	low	medium	low	low	high	low	low	very_low
medium	high	low	high	medium	medium	medium	low	medium	medium	medium
medium	high	low	high	medium	medium	medium	low	low	medium	high
low	medium	low	medium	medium	low	low	medium	low	medium	high
low	low	low	medium	medium	low	very_low	high	low	medium	medium
low	medium	medium	medium	low	high	medium	low	medium	medium	medium
low	medium	high	high	medium	low	medium	high	low	medium	medium
low	medium	low	high	low	medium	medium	high	low	medium	high
low	medium	high	low	medium	low	low	high	low	low	medium
low	high	low	high	medium	medium	medium	medium	medium	medium	high
medium	medium	high	medium	high	medium	high	medium	low	low	Very_high
medium	low	high	medium	medium	medium	medium	medium	low	high	high
medium	low	medium	medium	medium	low	low	low	low	low	medium
high	medium	medium	medium	high	high	high	low	medium	low	medium
high	high	medium	medium	medium	high	high	low	medium	low	medium
high	high	low	medium	medium	medium	medium	medium	medium	medium	high
high	high	low	high	medium	medium	medium	low	medium	medium	high
low	low	medium	medium	medium	low	low	low	low	low	high

Table 4.11 Actual Data of 86 sets Obtained from RSP Converted to Very_High, High, Medium, low , very low (2of 3)

low	low	high	medium	medium	low	medium	medium	low	medium	very_high
low	low	medium	medium	medium	low	low	high	low	high	high
high	high	high	medium	medium	high	high	low	medium	high	very_high
medium	high	high	low	high	low	very_low	low	low	medium	high
low	medium	high	low	medium	low	low	low	low	medium	medium
low	medium	high	medium	medium	low	very_low	low	low	medium	medium
low	medium	high	low	medium	low	very_low	low	low	high	high
low	medium	high	low	low	medium	medium	medium	low	medium	low
low	medium	high	medium	medium	low	medium	medium	low	medium	medium
low	high	high	low	medium	low	low	high	low	medium	high
high	high	high	low	medium	medium	medium	medium	medium	high	medium
high	high	high	medium	medium	medium	medium	medium	medium	low	high
high	high	medium	medium	medium	medium	high	low	low	high	high
high	medium	medium	medium	medium	low	low	low	low	low	very_high
medium	low	high	low	medium	high	very_high	low	medium	high	medium
medium	low	high	medium	medium	low	low	low	medium	high	very_high
medium	low	high	medium	medium	high	low	medium	medium	low	medium
medium	low	high	medium	high	low	medium	low	medium	high	very_low
medium	low	high	high	high	high	very_high	low	medium	low	medium
low	low	high	high	high	high	very_high	low	medium	medium	medium
low	low	high	high	high	high	very_high	medium	low	medium	low
medium	low	high	medium	medium	medium	medium	medium	medium	low	high
medium	medium	medium	medium	medium	medium	high	high	medium	low	medium
medium	medium	low	medium	medium	medium	medium	medium	medium	low	very_low
medium	medium	medium	medium	medium	high	medium	high	medium	low	high
medium	medium	low	high	medium	medium	medium	medium	low	low	high
medium	medium	medium	high	low	high	high	high	low	low	high
high	medium	low	high	medium	high	high	medium	low	low	medium
high	high	low	high	medium	high	medium	low	low	low	medium
high	medium	low	medium	medium	medium	low	low	medium	medium	medium
high	medium	low	medium	low	medium	medium	medium	medium	medium	medium
high	medium	low	medium	low	medium	medium	low	low	high	low
medium	medium	low	high	low	medium	medium	low	low	medium	high

Table 4.11 Actual Data of 86 sets Obtained from RSP Converted to Very_High, High, Medium, low , very low (3of 3)

RULE BASE FOR EXPERT BASED DATA OBTAINED USING ID3 PROGRAM

```
ghdt = high and   rt = high and      cti = high     ==> class = very_high
                                     cti = medium   ==> class = very_high
                                     cti = low      ==> class = high

                  rt = medium and    cti = high     ==> class = high
                                     cti = medium   ==> class = high
                                     cti = low      ==> class = medium

                  rt = low and       cti = high ==>     class = medium
                                     cti = medium ==> class = medium
                                     cti = low ==>      class = low


ghdt = medium and rt = high and      cti = high ==>     class = high
                                     cti = medium ==> class = high
                                     cti = low ==>      class = medium

                  rt = medium ==>           class = medium

                  rt = low and       cti = high ==>     class = medium
                                     cti = medium ==> class = low
                                     cti = low ==>      class = low


ghdt = low and    rt = high and      cti = high ==>     class = medium
                                     cti = medium ==> class = medium
                                     cti = low ==> class = low

                  rt = medium and    cti = high ==> class = low
                                     cti = medium ==> class = low
                                     cti = low ==> class = very_low

                  rt = low ==>       class = very_low
```

Figure 4.5: Decision Tree for Grain Temperature of Table 4.2

gt = very_high ==> class = very_high
gt = high and mixer_temp = high ==> class = high
mixer_temp = medium ==> class = high
mixer_temp = low and mix_time = high ==> class = high
mix_time = medium ==> class = high
mix_time = low ==> class = medium

gt = medium and mixer_temp = high ==> class = medium
mixer_temp = medium ==> class = medium
mixer_temp = low and mix_time = high ==> class = low
mix_time = medium ==> class = low
mix_time = low ==> class = medium

gt = low and mixer_temp = high and mix_time = high ==> class = medium
mix_time = medium ==> class = low
mix_time = low ==> class = low

mixer_temp = medium ==> class = low
mixer_temp = low ==> class = low

gt = very_low and mixer_temp = high and mix_time = high ==> class = low
mix_time = medium ==> class = low
mix_time = low ==> class = very_low

mixer_temp = medium and mix_time = high ==> class = low
mix_time = medium ==> class =
low

mix_time = low ==> class =
very_low

mixer_temp = low ==> class = very_low

Figure 4.6: Decision Tree for Mix Temperature of table 4.3

```
mix_temp = very_high and        rolls = high and    st = high ==> class = low
                                                    st = medium ==> class = low
                                                    st = low ==> class = medium

                                rolls = medium and          st = high ==> class = medium
                                                            st = medium ==> class = medium
                                                            st = low ==> class = high

                                rolls = low and     st = high ==> class = high
                                                    st = medium ==> class = high
                                                    st = low ==> class = very_high

mix_temp = high and      rolls = high ==> class = medium
                         rolls = medium and        st = high ==> class = medium
                                                   st = medium ==> class = medium
                                                   st = low ==> class = high

                         rolls = low ==> class = high

mix_temp = medium and    rolls = high ==> class = low
                         rolls = medium ==> class = low
                         rolls = low ==> class = medium

mix_temp = low ==> class = low
mix_temp = very_low and rolls = high ==> class = very_low
                        rolls = medium ==> class = very_low
                        rolls = low and    st = high ==> class = very_low
                                           st = medium ==> class = low
                                           st = low ==> class = low
```

Figure 4.7: Decision Tree for Press Mix Temperature of table 4.4

graphite = very_low and pmt = high and fp = high and pitch = high ==> class = very_high
pitch = medium ==> class = very_high
pitch = low ==> class = high

fp = medium and pitch = high ==> class = very_high
pitch = medium ==> class = very_high
pitch = low ==> class = high

fp = low and pitch = high ==> class = high
pitch = medium ==> class = high
pitch = low ==> class = medium

pmt = medium and fp = high ==> class = high
fp = medium and pitch = high ==> class = high
pitch = medium ==> class = high
pitch = low ==> class = medium

fp = low and pitch = high ==> class = high
pitch = medium ==> class = medium
pitch = low ==> class = medium

pmt = low and pitch = high ==> class = medium
pitch = medium and fp = high ==> class = high
fp = medium ==> class = medium
fp = low ==> class = medium

graphite = very_high and fp = high and pmt = high and pitch = high ==> class = low
pitch = medium ==> class = low
pitch = low ==> class = very_low

pmt = medium and pitch = high ==> class = low
pitch = medium ==> class = very_low
pitch = low ==> class = very_low

pmt = low ==> class = very_low

fp = medium and pmt = high and pitch = high ==> class = low
pitch = medium ==> class = very_low
pitch = low ==> class = very_low

pmt = medium ==> class = very_low
pmt = low ==> class = very_low

fp = low ==> class = very_low

Figure 4.8: Decision Tree for Green Bulk Density of table 4.5(1 of 3)

```
graphite = high and      fp = high and     pmt = high and    pitch = high  ==> class = medium
                                                             pitch = medium  ==> class = low
                                                             pitch = low  ==> class = low

                                           pmt = medium  ==> class = low
                                           pmt = low  ==> class = low

                         fp = medium and pmt = high  ==> class = low
                                         pmt = medium  ==> class = low
                                         pmt = low  ==> class = very_low

                         fp = low and      pitch = high and pmt = high  ==> class = low
                                                            pmt = medium  ==> class = low
                                                            pmt = low  ==> class = very_low

                                           pitch = medium  ==> class = very_low
                                           pitch = low  ==> class = very_low


graphite = medium and pmt = high and fp = high  ==> class = medium
                                fp = medium and pitch = high  ==> class = medium
                                                pitch = medium  ==> class = medium
                                                pitch = low  ==> class = low

                                fp = low and    pitch = high  ==> class = medium
                                                pitch = medium  ==> class = low
                                                pitch = low  ==> class = low


                      pmt = medium and  fp = high  ==> class = medium
                                        fp = medium and pitch = high  ==> class = medium
                                                        pitch = medium  ==> class = medium
                                                        pitch = low  ==> class = low

                                        fp = low and    pitch = high  ==> class = medium
                                                        pitch = medium  ==> class = low
                                                        pitch = low  ==> class = low


                      pmt = low and     fp = high and   pitch = high  ==> class = low
                                                        pitch = medium  ==> class = medium
                                                        pitch = low  ==> class = low

                                        fp = medium  ==> class = low
                                        fp = low and    pitch = high  ==> class = low
                                                        pitch = medium  ==> class = low
                                                        pitch = low  ==> class = very_low
```

Figure 4.8: Decision Tree for Green Bulk Density of table 4.5(2 of 3)

graphite = low and pitch = high and
pmt = high and
fp = high ==> class = high
fp = medium ==> class = high
fp = low ==> class = medium

pmt = medium ==> class = high
pmt = low and
fp = high ==> class = medium
fp = medium ==> class = medium
fp = low ==> class = low

pitch = medium and
pmt = high and
fp = high ==> class = high
fp = medium ==> class = high
fp = low ==> class = medium

pmt = medium ==> class = medium
pmt = low ==> class = medium

pitch = low and
pmt = high and
fp = high ==> class = medium
fp = medium ==> class = medium
fp = low ==> class = low

pmt = medium ==> class = medium
pmt = low and
fp = high ==> class = medium
fp = medium ==> class = medium
fp = low ==> class = low

Figure 4.8: Decision Tree for Green Bulk Density of table 4.5(3 of 3)

gbd = medium and	pitch = high and	tempering = perfect ==> class = high
		tempering = under ==> class = high
		tempering = over ==> class = medium
	pitch = medium and	tempering = perfect ==> class = medium
		tempering = under ==> class = medium
		tempering = over ==> class = low
	pitch = low and	tempering = perfect ==> class = low
		tempering = under ==> class = medium
		tempering = over ==> class = low
gbd = low and	pitch = high and	tempering = perfect ==> class = medium
		tempering = under ==> class = medium
		tempering = over ==> class = low
	pitch = medium and	tempering = perfect ==> class = medium
		tempering = under ==> class = medium
		tempering = over ==> class = low
	pitch = low and	tempering = perfect ==> class = low
		tempering = under ==> class = low
		tempering = over ==> class = very_low
gbd = very_low and	tempering = perfect ==> class = very_low	
	tempering = under and	pitch = high ==> class = low
		pitch = medium ==> class = low
		pitch = low ==> class = very_low
	tempering = over ==> class = very_low	
gbd = very_high and	pitch = high and	tempering = perfect ==> class = very_high
		tempering = under ==> class = very_high
		tempering = over ==> class = high
	pitch = medium and	tempering = perfect ==> class = very_high
		tempering = under ==> class = high
		tempering = over ==> class = high
	pitch = low and	tempering = perfect ==> class = high
		tempering = under ==> class = medium
		tempering = over ==> class = very_high
gbd = high and	pitch = high and	tempering = perfect ==> class = very_high
		tempering = under ==> class = high
		tempering = over ==> class = high
	pitch = medium and	tempering = perfect ==> class = high
		tempering = under ==> class = medium
		tempering = over ==> class = medium
	pitch = low ==> class = medium	

Figure 4.9: Decision Tree for Cold Crushing Strength of table 4.6

gbd = very_high and pitch = high ==> class = medium
pitch = medium ==> class = low
pitch = low ==> class = low

gbd = high and pitch = high ==> class = high
pitch = medium ==> class = medium
pitch = low ==> class = low

gbd = medium and pitch = high ==> class = high
pitch = medium ==> class = medium
pitch = low ==> class = medium

gbd = low and pitch = high ==> class = high
pitch = medium ==> class = medium
pitch = low ==> class = medium

gbd = very_low ==> class = high

Figure 4.10: Decision Tree for Coked Porosity of table 4.7

```
mp = high and    ccs = high and    gbd = very_low and    rc = very_high ==> class = vv_high
                                                         rc = high ==> class = very_high
                                                         rc = medium ==> class = high

                                   gbd = high ==> class = very_high
                                   gbd = medium and      rc = high ==> class = vv_high
                                                         rc = medium ==> class = vv_high
                                                         rc = low ==> class = very_high

                                   gbd = low and         rc = very_high ==> class = vv_high
                                                         rc = high ==> class = vv_high
                                                         rc = medium ==> class = very_high
                                                         rc = low ==> class = very_high

                                   gbd = very_high ==> class = high

                 ccs = medium and gbd = very_low and     rc = very_high ==> class = very_high
                                                         rc = high ==> class = very_high
                                                         rc = medium ==> class = medium

                                   gbd = high ==> class = very_high
                                   gbd = medium and      rc = high ==> class = vv_high
                                                         rc = medium ==> class = vv_high
                                                         rc = low ==> class = very_high

                                   gbd = low and   rc = very_high ==> class = vv_high
                                                   rc = high ==> class = vv_high
                                                   rc = medium ==> class = very_high
                                                   rc = low ==> class = very_high

                                   gbd = very_high ==> class = high

                 ccs = low and     gbd = very_low and    rc = very_high ==> class = very_high
                                                         rc = high ==> class = very_high
                                                         rc = medium ==> class = medium

                                   gbd = high ==> class = high
                                   gbd = medium and      rc = high ==> class = vv_high
                                                         rc = medium ==> class = vv_high
                                                         rc = low ==> class = very_high

                                   gbd = low and   rc = very_high ==> class = vv_high
                                                   rc = high ==> class = very_high
                                                   rc = medium ==> class = very_high
                                                   rc = low ==> class = high

                                   gbd = very_high ==> class = medium
```

Figure 4.11: Decision Tree for HMOR of table 4.8 (1 of 3)

mp = medium and ccs high and gbd = very_low and rc = very_high ==> class = high
rc = high ==> class = high
rc = medium ==> class = medium

gbd = high ==> class = high
gbd = medium and rc = high ==> class = vv_high
rc = medium ==> class = vv_high
rc = low ==> class = high

gbd = low and rc = very_high ==> class = vv_high
rc = high ==> class = very_high
rc = medium ==> class = very_high
rc = low ==> class = high

gbd = very_high ==> class = medium

ccs = medium and gbd = very_low and rc = very_high ==> class = high
rc = high ==> class = high
rc = medium ==> class = medium

gbd = high ==> class = high
gbd = medium and rc = high ==> class = vv_high
rc = medium ==> class = very_high
rc = low ==> class = high

gbd = low and rc = very_high ==> class = vv_high
rc = high ==> class = very_high
rc = medium ==> class = very_high
rc = low ==> class = high

gbd = very_high ==> class = medium

ccs = low and gbd = very_low and rc = very_high ==> class = high
rc = high ==> class = medium
rc = medium ==> class = low

gbd = high ==> class = high
gbd = medium and rc = high ==> class = very_high
rc = medium ==> class = high
rc = low ==> class = high

gbd = low and rc = very_high ==> class = very_high
rc = high ==> class = high
rc = medium ==> class = high
rc = low ==> class = medium

gbd = very_high ==> class = medium

Figure 4.11: Decision Tree for HMOR of table 4.8 (2 of 3)

mp = low and rc = very_high and gbd = low and ccs = high ==> class = high
ccs = medium ==> class = high
ccs = low ==> class = medium

gbd = very_low ==> class = medium

rc = high and ccs = high and gbd = medium ==> class = high
gbd = low ==> class = high
gbd = very_low ==> class = medium

ccs = medium ==> class = medium
ccs = low ==> class = medium

rc = medium and gbd = medium and ccs = high ==> class = medium
ccs = medium ==> class = medium
ccs = low ==> class = low

gbd = low and ccs = high ==> class = medium
ccs = medium ==> class = medium
ccs = low ==> class = low

gbd = very_low and ccs = high ==> class = low
ccs = medium ==> class = very_low
ccs = low ==> class = very_low

rc = low and gbd = high and ccs = high ==> class = medium
ccs = medium ==> class = medium
ccs = low ==> class = low

gbd = medium and ccs = high ==> class = medium
ccs = medium ==> class = low
ccs = low ==> class = low

gbd = low and ccs = high ==> class = low
ccs = medium ==> class = very_low
ccs = low ==> class = very_low

rc = very_low ==> class = low

Figure 4.11: Decision Tree for HMOR of table 4.8(3 of 3)

```
chp = high and   cp = high and   rc = medium  ==> class = medium
                                 rc = low  ==> class = low
                                 rc = very_low  ==> class = low
                                 rc = very_high  ==> class = high
                                 rc = high  ==> class = high

                 cp = medium and rc = medium  ==> class = high
                                 rc = low  ==> class = medium
                                 rc = very_low  ==> class = medium
                                 rc = very_high  ==> class = very_high
                                 rc = high  ==> class = high

                 cp = low and    rc = medium  ==> class = high
                                 rc = low  ==> class = medium
                                 rc = very_low  ==> class = medium
                                 rc = very_high  ==> class = very_high
                                 rc = high  ==> class = very_high

chp = medium and rc = medium and  cp = high  ==> class = medium
                                  cp = medium  ==> class = medium
                                  cp = low  ==> class = high

                 rc = low and     cp = high  ==> class = low
                                  cp = medium  ==> class = medium
                                  cp = low  ==> class = medium

                 rc = very_low and cp = high  ==> class = very_low
                                  cp = medium  ==> class = low
                                  cp = low  ==> class = low

                 rc = very_high and cp = high  ==> class = high
                                  cp = medium  ==> class = high
                                  cp = low  ==> class = very_high

                 rc = high and    cp = high  ==> class = high
                                  cp = medium  ==> class = high
                                  cp = low  ==> class = very_high

chp = low and  rc = medium  ==> class = low
               rc = low and    cp = high  ==> class = very_low
                               cp = medium  ==> class = very_low
                               cp = low  ==> class = low

               rc = very_low and cp = high  ==> class = very_low
                               cp = medium  ==> class = very_low
                               cp = low  ==> class = low

               rc = very_high and cp = high  ==> class = low
                               cp = medium  ==> class = medium
                               cp = low  ==> class = medium

               rc = high and   cp = high  ==> class = low
                               cp = medium  ==> class = low
                               cp = low  ==> class = medium
```

Figure 4.12: Decision Tree for Slag Corrosion Resistance of table 4.9

```
rc = very_low and  mp = high and      gbd = very_high ==> class = medium
                                      gbd = high ==> class = low
                                      gbd = medium ==> class = low
                                      gbd = low ==> class = low

                   mp = medium and gbd = very_high ==> class = low
                                      gbd = high ==> class = very_low
                                      gbd = medium ==> class = low
                                      gbd = low ==> class = very_low

                   mp = low ==> class = very_low

rc = low and       gbd = high and     mp = high ==> class = medium
                                      mp = medium ==> class = low
                                      mp = low ==> class = low

                   gbd = medium and mp = high ==> class = medium
                                      mp = medium ==> class = low
                                      mp = low ==> class = low

                   gbd = low and      mp = high ==> class = low
                                      mp = medium ==> class = very_low
                                      mp = low ==> class = very_low

rc = high and      gbd = medium and mp = high ==> class = very_high
                                      mp = medium ==> class = high
                                      mp = low ==> class = medium

                   gbd = low and      mp = high ==> class = high
                                      mp = medium ==> class = high
                                      mp = low ==> class = medium

                   gbd = very_low and         mp = high ==> class = medium
                                              mp = medium ==> class = medium
                                              mp = low ==> class = low

rc = medium and    gbd = medium and  mp = high ==> class = high
                                      mp = medium ==> class = high
                                      mp = low ==> class = medium

                   gbd = low and      mp = high ==> class = medium
                                      mp = medium ==> class = low
                                      mp = low ==> class = low

                   gbd = very_low and mp = high ==> class = medium
                                      mp = medium ==> class = low
                                      mp = low ==> class = low

rc = very_high and gbd = low and      mp = high ==> class = very_high
                                      mp = medium ==> class = very_high
                                      mp = low ==> class = high

                   gbd = very_low and mp = high ==> class = very_high
                                      mp = medium ==> class = high
                                      mp = low ==> class = medium
```

Figure 4.13: Decision Tree for Oxidation Resistance of table 4.10

Fig 4.14 Tree for data set of 86 samples shown on two pages(1 of 2)

```
MTM = H & PMT = L ==> CLASS = M
          PMT = H & GHDT = M & MT = M & FP = L & CTI = M ==> CLASS = VH
                                                 CTI = H ==> CLASS = M
          PMT = VL & CTI = M & GHDT = M ==> CLASS = VL
                               GHDT = H & MT = L & MIX = L & ST = L & ROLLS = L & FP = M & RT = L ==> CLASS = M
                                                                                            RT = H ==> CLASS = H
          PMT = M & GHDT = M ==> CLASS = M
                    GHDT = H ==> CLASS = M
                    GHDT = L ==> CLASS = VL

          PMT = VH & ST = L ==> CLASS = M
                     ST = M ==> CLASS = L

MTM = M & MIX = L & ROLLS = L & CTI = M ==> CLASS = M

                                CTI = L & RT = L & GHDT = M & MT = L ==> CLASS = VL
                                                              MT = M ==> CLASS = H
                                                 GHDT = L ==> CLASS = M

                                          RT = M & MT = L ==> CLASS = L
                                                   MT = M ==> CLASS = H

                                          RT = H & GHDT = L & PMT = M & ST = M & MT = L ==> CLASS = M
                                                                                MT = M ==> CLASS = VH
                                                   GHDT = M & FP = L ==> CLASS = M
                                                              FP = H ==> CLASS = H
                                                              FP = M ==> CLASS = M
                                                   GHDT = H ==> CLASS = H

                               CTI = H & MT = M & FP = L & GHDT = L ==> CLASS = L
                                                           GHDT = M ==> CLASS = VH

                  ROLLS = M & GHDT = L & MT = M & CTI = L ==> CLASS = L
                                                  CTI = M ==> CLASS = VH
         MIX = H & FP = M ==> CLASS = VL
                  FP = H & ROLLS = L ==> CLASS = M
                           ROLLS = M & CTI = H & RT = L ==> CLASS = M
                                                 RT = H ==> CLASS = VH
                                       CTI = M ==> CLASS = M

                  FP = L & CTI = M & MT = M & ROLLS = M & GHDT = L ==> CLASS = M
                                                          GHDT = M ==> CLASS = H
                           CTI = H ==> CLASS = M

         MIX = M & GHDT = H & ROLLS = M & PMT = L ==> CLASS = M
                                          PMT = M & RT = L & MT = M ==> CLASS = H
                                                             MT = H & ST = M ==> CLASS = H
                                                                      ST = L & FP = M & CTI = M ==> CLASS = M
                                                                                        CTI = H ==> CLASS = H
                                                    RT = H & CTI = H & ST = M & MT = L ==> CLASS = M
                                                                                MT = M ==> CLASS = H
                               ROLLS = L & ST = M ==> CLASS = M
                                           ST = H ==> CLASS = L
                                           ST = L ==> CLASS = H

                  GHDT = L & ST = L ==> CLASS = H
                             ST = M & PMT = L ==> CLASS = L
                                      PMT = M ==> CLASS = H

                  GHDT = M & ROLLS = L ==> CLASS = H
                             ROLLS = M & CTI = M & RT = M ==> CLASS = M
                                                   RT = L ==> CLASS = VL
                                         CTI = H & MT = H ==> CLASS = M
                                                   MT = M & RT = L & PMT = M ==> CLASS = L
                                                                     PMT = L ==> CLASS = M
```

Fig 4.14 Tree for data set of 86 samples shown on two pages(2 of 2)

```
MTM = L & ROLLS = M & PMT = H & MIX = H & FP = M & CTI = H ==> CLASS = L
                                                   CTI = L ==> CLASS = H
                              PMT = M ==> CLASS = M


            ROLLS = L & PMT = H & CTI = H & RT = M ==> CLASS = M
                                            RT = L & GHDT = M & MT = L & MIX = M & ST = M & FP = M ==> CLASS = M
                                                                                           FP = M ==> CLASS = L

                                   CTI = M ==> CLASS = H

                              PMT = M & MIX = L ==> CLASS = M
                                        MIX = M & GHDT = L ==> CLASS = M
                                                  GHDT = M & RT = H ==> CLASS = L
                                                             RT = L & MT = M ==> CLASS = L
                                                                      MT = H ==> CLASS = H
```

PLANT DATA FOR PROCESS UP TO GBD

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLL	FP	GBD
414	519	6	126	16	153	132	85	1	1484	3.09
385	532	5	128	15	135	128	90	1	1353	3.07
380	480	6	123	16	140	130	28	1	1345	3.11
361	494	6	126	16	135	128	108	0	1345	3.05
403	511	6	120	14	145	138	45	0	1484	3.06
364	502	6	124	14	125	121	20	0	1345	3.05
353	454	6	120	14	125	120	20	0	1484	3.07
365	466	6	116	14	140	125	60	0	1396	3.05
360	411	7	123	14	144	144	10	1	1476	3.09
365	381	9	120	12	137	130	50	1	1360	3.06
368	369	7	118	14	142	130	85	0	1484	3.06
380	490	7	105	16	135	135	20	1	1484	3.08
364	492	6	135	15	153	140	40	0	1491	3.07
359	484	7	133	16	143	135	210	0	1462	3.12
404	505	6	127	16	136	135	12	0	1258	3.14
374	521	6	129	17	135	134	19	0	1353	3.1
354	500	6	129	17	132	131	12	0	1360	3.13
364	539	6	129	16	134	132	28	0	1338	3.12
359	539	6	131	14	138	135	31	0	1353	3.12
373	587	6	129	14	132	128	128	1	1345	3.12
460	380	7	125	16	145	135	30	1	1491	3.12
357	466	8	131	16	146	136	38	1	1484	3.06
349	475	8	137	16	146	138	83	1	1476	3.11
346	472	8	135	20	145	140	90	2	1484	3.05
382	528	6	118	18	160	140	45	1	1476	3.15
376	498	4	125	17	129	125	5	1	1345	3.08
395	490	7	123	28	135	130	5	1	1353	3.08
377	510	7	129.6	26	140	131	20	1	1345	3.04
363	479	7	135	27	142	132	30	1	1345	3.08
355	477	7	136	27	140	130	40	1	1353	3.06
330	614	5	120	7	120	115	76	1	1411	3.1

Table 4.12 31 Data Sample Points of RSP Data

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLLS	FP	FP	GBD
319.5	574.5	5	129	8	134	132	55	0	195	1418.18	3.1
248.6	351.5	3	122	8	124	105	90	1	195	1418.18	3.06
359	485	6	107	14	115	112	30	0	190	1381.82	3.01
276	420.1	7	125	11	131	122	35	1	204	1483.64	3.13
272	445	7	126	13	126	120	175	0	210	1527.27	3.11
386	599	4	127	14	135	122	48	1	212	1541.82	3.11
361	570	4	130	9	130	120	108	0	200	1454.55	3.08
388.8	435.8	5	127	31.5	128	115	20	1	193	1403.64	3.11
370	428	7	130	33	148	122	65	1	193	1403.64	3.04
423	500	5.8	127	7	128	128	31	0	211	1534.55	3.09
440	579	5.3	104	8	135	115	26	1	194	1410.91	3.12
438	539	4	106	10	140	128	20	1	212	1541.82	3.09
416	494	4	128	11	135	125	30	1	210	1527.27	3.09
416	514	4	131	11	140	127	45	1	211	1534.55	3.09
380	440	4	123	11	135	123	45	1	210	1527.27	3.08
309	340	5	105	11	130	122	33	0	214	1556.36	3.14
286	314	6	108	8	130	117	42	1	211	1534.55	3.06
265	290	7	105	8	120	120	43	0	212	1541.82	3.11
288	301	6	105	9	120	120	38	0	209	1520	3.08
518	489	5	118	8	140	128	115	0	195	1418.18	3.02
454	389	5	132	5	150	130	135	1	194	1410.91	3.06
403	516	4	106	7	133	129	50	0	194	1410.91	3.06
395	499	4	111	7	132	127	75	0	195	1418.18	3.09
388	395	5	120	11	125	125	35	0	193	1403.64	3.08
376	420	6	124	9	130	123	25	1	195	1418.18	3.13
408	405	6	128	5	127	126	35	0	195	1418.18	3.11
392	407	5	128	5	122	122	25	0	197	1432.73	3.12
384	415	5	122	5	125	125	40	0	196	1425.45	3.11
371.2	434.3	7	120	8	140	124	175	0	210	1527.27	3.1
350	717	5	113	14	112	112	20	0	195	1418.18	3.09
393.3	437	4	125	20	144	125	170	0	195	1418.18	3.03
555.4	436.3	5	134	10	178	127	180	1	194	1410.91	3.05
255	398	10	115	8	128	128	21	0	195	1418.18	3.14
287	560	8.5	115	6.25	166	129	25	1	195	1418.18	3.13
307	630	6	119	7.5	124	124	32	0	193	1403.64	3.11
293	505	5	114	9	116	116	80	0	193	1403.64	3.05
345.7	685.9	4.5	126	8	130	125	10	1	195	1418.18	3.1
333.8	658	4	126	11	128	125	20	0	196	1425.45	3.15
276	453.5	4.5	123	8	118	118	40	0	194	1410.91	3.14
248.7	417.1	5	121	8	110	109	186	0	197	1432.73	3.1
304	517	6	120	6	145	125	25	1	194	1410.91	3.11
286	493	8	126	8	125	124	185	0	196	1425.45	3.09
291	513	5	124	5	135	125	145	0	195	1418.18	3.13

284	506.6	7	91	8.5	123	114	180	0	193	1403.64	3.12
314	717	5	130	9	135	120	47	1	194	1410.91	3.13
340	441	8	119	18	135	130	37	0	185	1345.45	3.17
321.2	425.1	8	124	8	130	125	31	0	218	1585.45	3.15
330	408	6	122	12	116	114	20	0	188	1367.27	3.12
380	520	6	118	16	140	128	13	1	192	1396.36	3.11
417	602	6	118	9	143	130	25	1	192	1396.36	3.1
401	580	5	124	8	134	120	43	1	195	1418.18	3.15
389	567	5	125	11	135	122	17	1	196	1425.45	3.14
320	410	6	117	8	120	118	27	0	192	1396.36	3.16
292	370	7	119	11	125	121	50	0	199	1447.27	3.18
285	366	6	119	10	120	114	78	0	202	1469.09	3.15
396	536	8	124	12	157	130	25	1	203	1476.36	3.17
354	552	7	112	14	115	110	20	0	201	1461.82	3.15
272	482	9	116	10	115	114	6	0	198	1440	3.11
278	480	7	117	9	113	111	13	0	201	1461.82	3.09
271	472	7	115	12	115	112	26	0	203	1476.36	3.13
296	524	9	114	7	127	125	29	0	194	1410.91	3.08
285	509	9	120	10	125	124	45	0	201	1461.82	3.09
303	530	8	115	9	122	118	78	0	199	1447.27	3.13
418	556	7	113	8	135	125	30	1	204	1483.64	3.1
400	544	7	117	8	137	126	30	1	185	1345.45	3.16
396	548	6	119	8	127	127	15	0	204	1483.64	3.13
380	490	6	121	9	119	119	10	0	183	1330.91	3.17
338	418	8	106	10	148	142	15	1	204	1483.64	3.12
329	428	7	124	11	125	115	10	1	204	1483.64	3.17
344	428	7	117	11	140	118	57	1	185	1345.45	3.12
344	421	8	121	17	124	120	17	1	204	1483.64	3.05
323	401	9	126	18	150	137	5	1	191	1389.09	3.09
307	392	8	130	19	148	142	13	1	199	1447.27	3.11
308	410	8	130	19	143	135	75	0	195	1418.18	3.08
323	392	7	118	9	136	125	67	1	190	1381.82	3.14
334	467	6	120	8	137	127	85	1	190	1381.82	3.09
376	505	5	117	8	135	124	72	1	190	1381.82	3.03
356	487	6	123	8	138	125	79	1	190	1381.82	3.13
368	494	5	125	10	136	124	72	0	190	1381.82	3.14
377	507	6	127	7	140	128	91	0	190	1381.82	3.13
390	501	5	125	9	142	129	62	0	190	1381.82	3.12
383	583	5	128	8	138	124	25	0	190	1381.82	3.12
391	506	5	122	11	130	118	15	1	195	1418.18	3.1
394	522	4	124	5	137	121	30	1	198	1440	3.11
389	516	5	121	6	127	125	5	0	203	1476.36	3.06
375	501	4	126	4	128	126	5	0	196	1425.45	3.13

Table (4.13) Actual Data of 86 sets Obtained from RSP

Appendix D

LISTING OF ALL RULES AUTOMATICALLY GENERATED BY CART

TREEE RULES

1. If (ST <= 58.5 && GHDT <= 537.5 && CTI <= 342 && MIXING TIME <= 10.5 && MIXER TEMPERATURE <= 114.6) {Class = low}

2. If (ST <= 58.5 &&GHDT <= 537.5 &&CTI <= 342 &&MIXING TIME <= 10.5 && MIXER TEMPERATURE > 114.6) {Class = high}

3. If (ST <= 58.5 &&GHDT <= 537.5 &&CTI <= 342 &&MIXING TIME > 10.5) {Class = very high}

4. If (ST <= 58.5 && CTI > 342 &&GHDT <= 532 && FP <= 1337.5){Class = very high}

5. If (ST <= 58.5 &&CTI > 342 &&GHDT <= 532 &&FP > 1337.5 &&PMT <= 121.5) {Class = very low}

6. If (GHDT <= 532 &&FP > 1337.5&&PMT > 121.5 &&CTI > 342&&CTI <= 405.5&& ST <= 29 &&MIXER TEMPERATURE <= 129.3 &&MIX_TEMP <= 137.5) {Class = low}

7. If (GHDT <= 532 &&FP >1337.5 &&PMT > 121.5 &&CTI > 342 &&CTI <= 405.5 && ST <= 29 &&MIXER TEMPERATURE <= 129.3 &&MIX_TEMP > 137.5) {Class = medium}

8. If (GHDT <= 532 &&FP > 1337.5 &&PMT > 121.5 &&CTI > 342 &&CTI<=405.5 && ST <= 29 &&MIXER TEMPERATURE > 129.3) {Class = very low}

9. If (FP > 1337.5 &&PMT > 121.5 &&CTI > 342 &&CTI<= 405.5 &&ST > 29 &&ST <= 58.5 && GHDT <= 522 &&MIXING TIME <= 6) {Class = medium}

10. If (FP > 1337.5 &&PMT > 121.5 &&CTI > 342 &&CTI <= 405.5 &&ST > 29 &&ST <= 58.5 &&GHDT <= 522 &&MIXING TIME > 6) {Class = low}

11. If (FP > 1337.5 &&PMT > 121.5 &&CTI > 342 &&CTI <= 405.5 &&ST > 29 &&ST <= 58.5 &&GHDT > 522 &&GHDT <= 532) {Class = high}

12. If (ST <= 58.5 &&GHDT <= 532 &&FP > 1337.5 &&PMT > 121.5 &&CTI > 405.5) {Class = medium}

13. If (ST <= 58.5 &&CTI > 342 &&GHDT > 532 &&GHDT <= 537.5){Class = very high}

14. If(ST <= 58.5 &&CTI <= 409 &&PMT <= 130.5 &&GHDT > 537.5 &&GHDT <=81.5)
{Class = high}

15. If (ST <= 58.5 &&CTI <= 409 &&PMT <= 130.5 &&GHDT > 581.5){Class = medium}

16. If (ST <= 58.5 &&GHDT > 537.5 &&CTI <= 409 &&PMT > 130.5){Class = medium}

17. If (ST <= 58.5 &&GHDT > 537.5 && CTI > 409){Class = medium}

18. If (ST > 58.5 &&GHDT <= 413.55) {Class = low}

19. If (ST > 58.5 &&GHDT > 413.55 &&GHDT <= 505.8 &&CTI <= 289.5)
{Class = medium}

20. If (ST > 58.5 &&GHDT > 413.55 &&GHDT <= 505.8 &&CTI > 289.5)
{Class = very low}

21. If (ST > 58.5 && GHDT > 505.8) {Class = high}

OUTPUTS OBTAINED USING CART ANALYSIS

Variable Importance

The variable importance as analyzed by Cart is shown below,

VARIABLE	% IMPORTANCE
GHDT	100.000
MIX_TEMP	89.650
CTI	77.924
MIXER_TEMP	60.601
ST	55.449
PMT	54.715
MIXING_TIME	54.137
RT	47.867
FP	30.409
NO_OF_ROLL	11.518

However as per the expert opinion the order of variable importance should have been as under.

- PMT
- MIX TEMP
- MIXING TIME
- CTI
- GHDT
- RT
- ST
- FP
- NO of ROLLS

Gain Charts

The gain charts represent the percentage of a class in the population as it increases from 0 % to 100 %. This is useful in the sense that one gets the overall picture of how many instances of each class gets included as the population increases to 100%. Here Class 1 to class 4 represents GBD Very Low, Low, Medium, High .

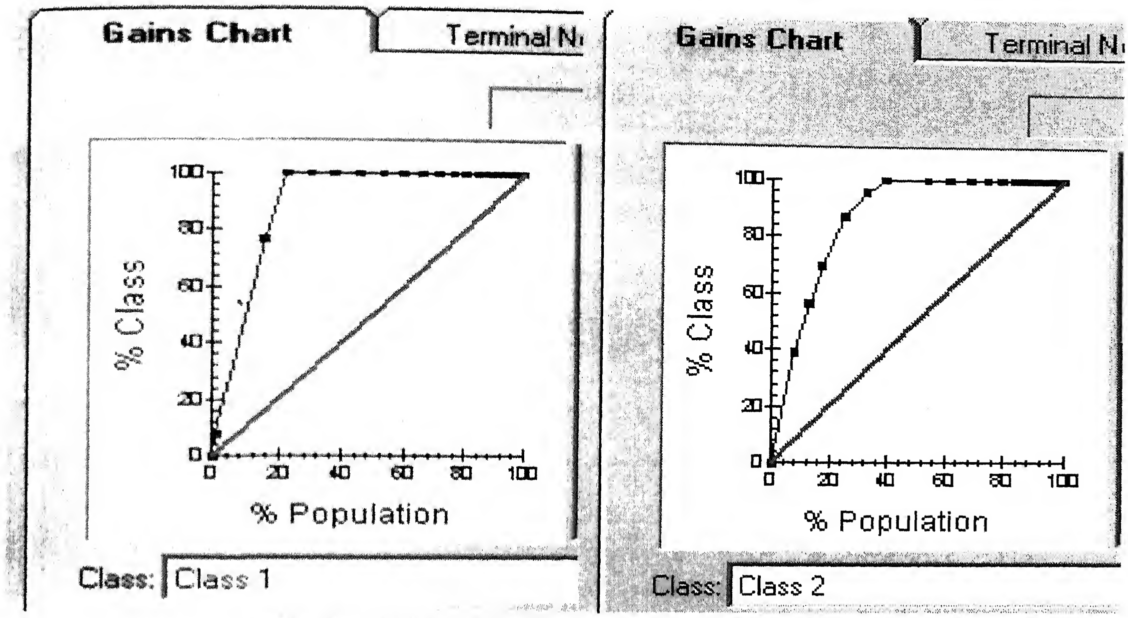


Fig 5.7 Gain Chart Class1 & Class 2 (VL & L)

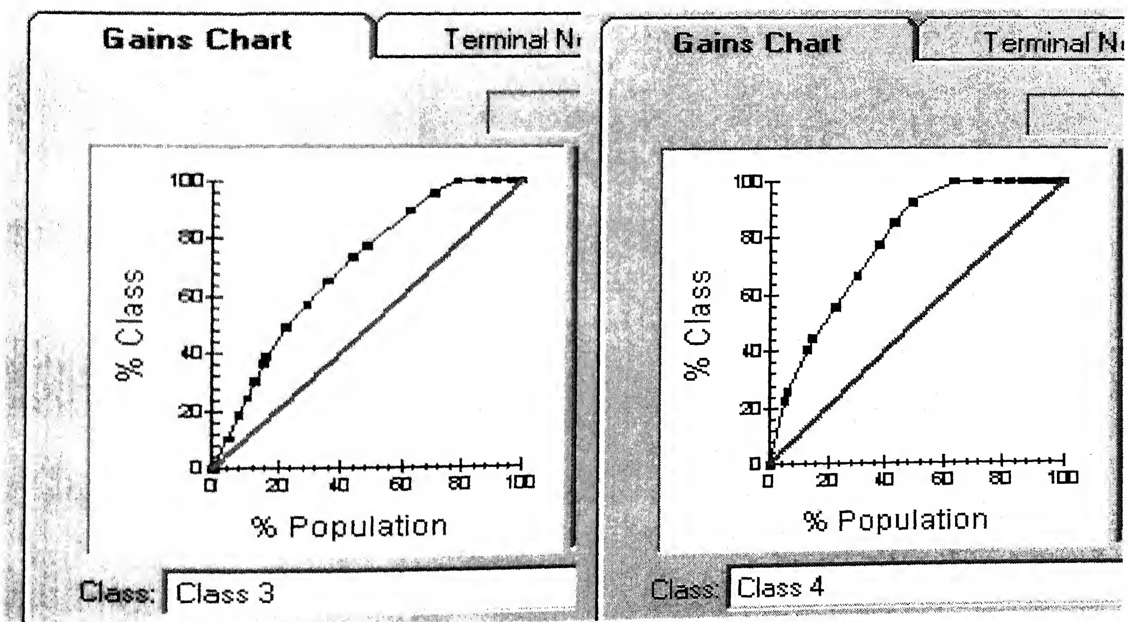


Fig 5.7 Gain Chart Class3 & Class 4 (M & H)

Composition of Terminal Nodes

For a total of 21 terminal Nodes generated by the Cart tree we have 21 Rules sets one for each terminal node. The % Class composition of these nodes based on these 21 rules is displayed by Bar Charts and Pie charts below for Node 1 to Node 6. Here the classes 1 to 6 (VL, L, M, H, VH) are represented by colour codes.

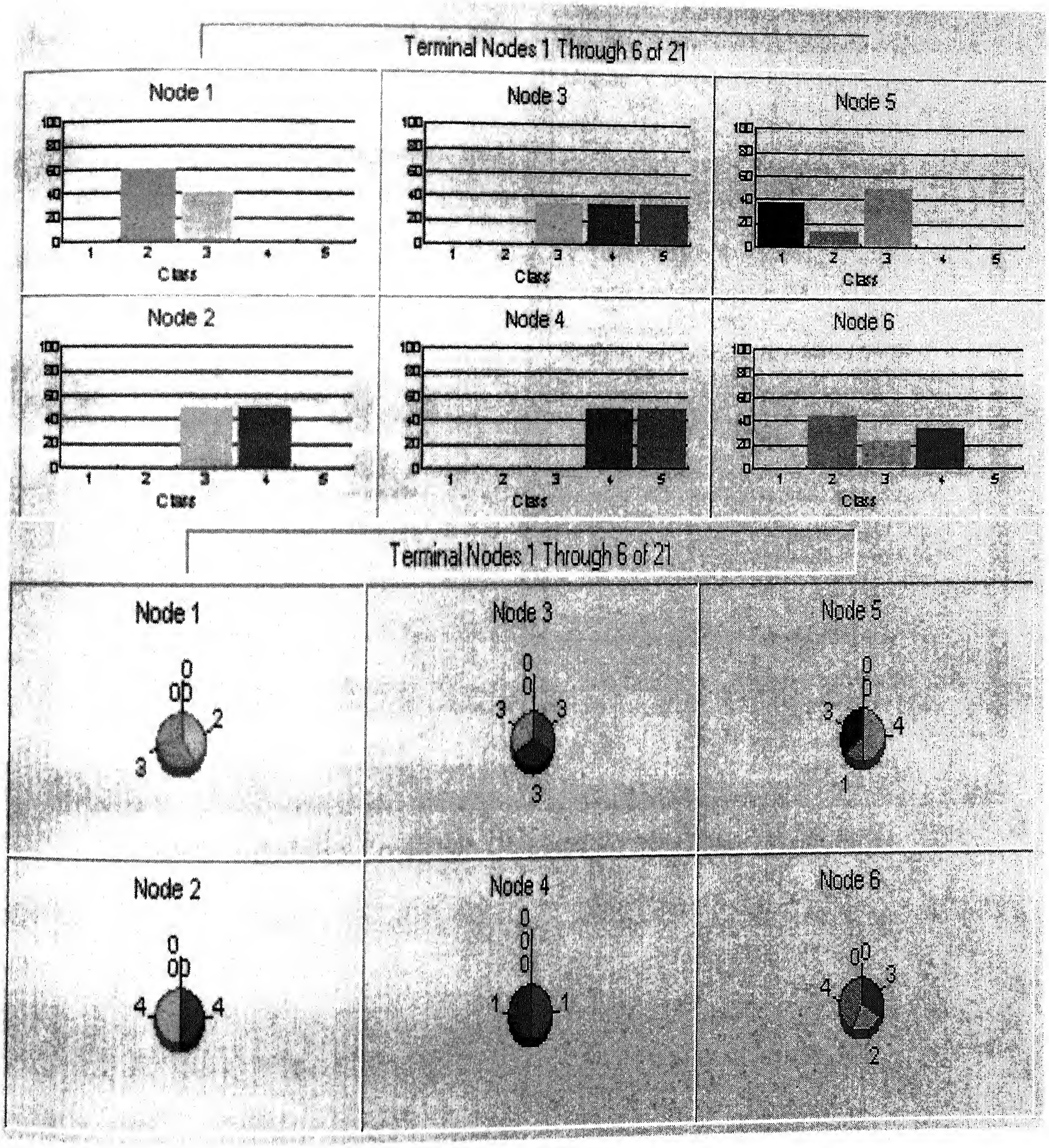


Fig 5.8 Bar Charts and Pie Charts for % Class composition of Nodes 1 to Node 6

PROBABILTY PREDICTION FOR GBD TAKING A SPECIFIC EXAMPLE

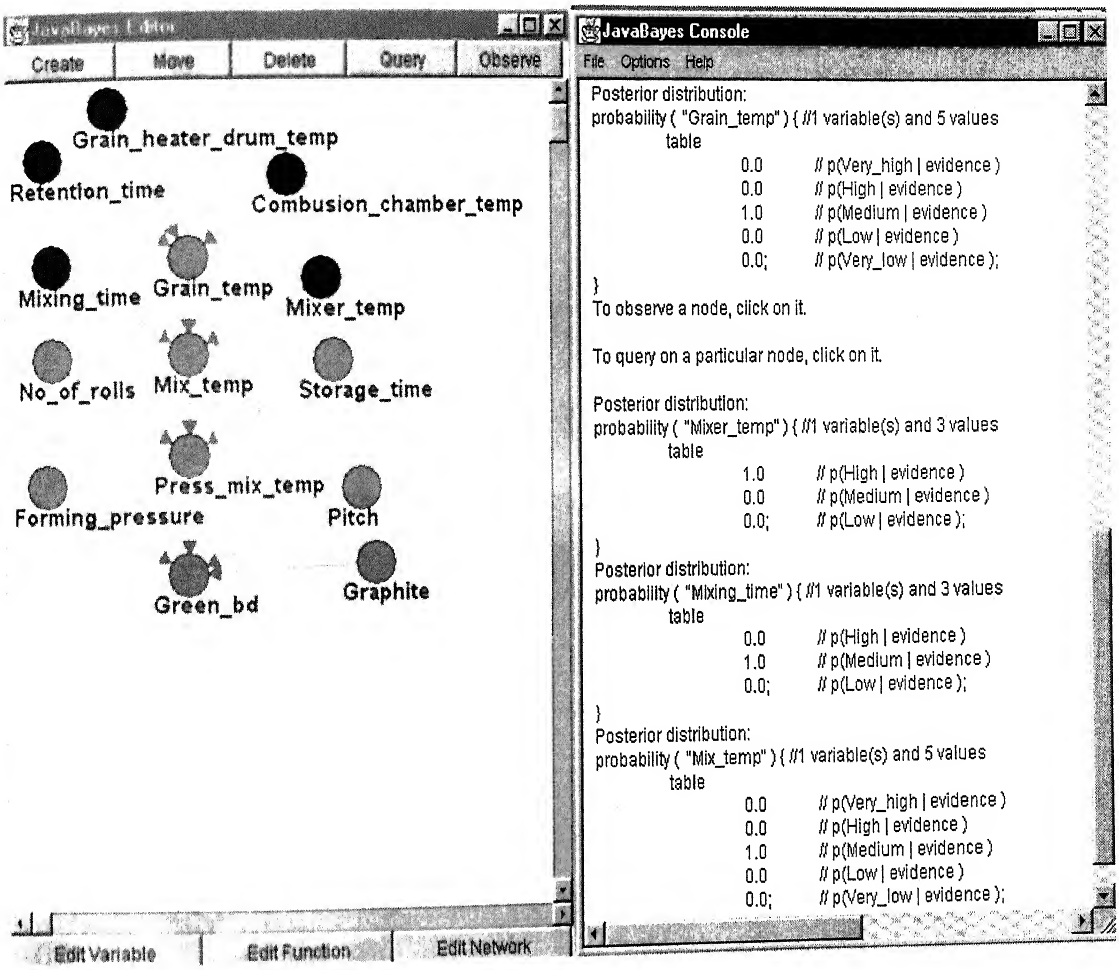


Fig 6.6 Probability Prediction Obtained for Mix Temp (Medium=1)

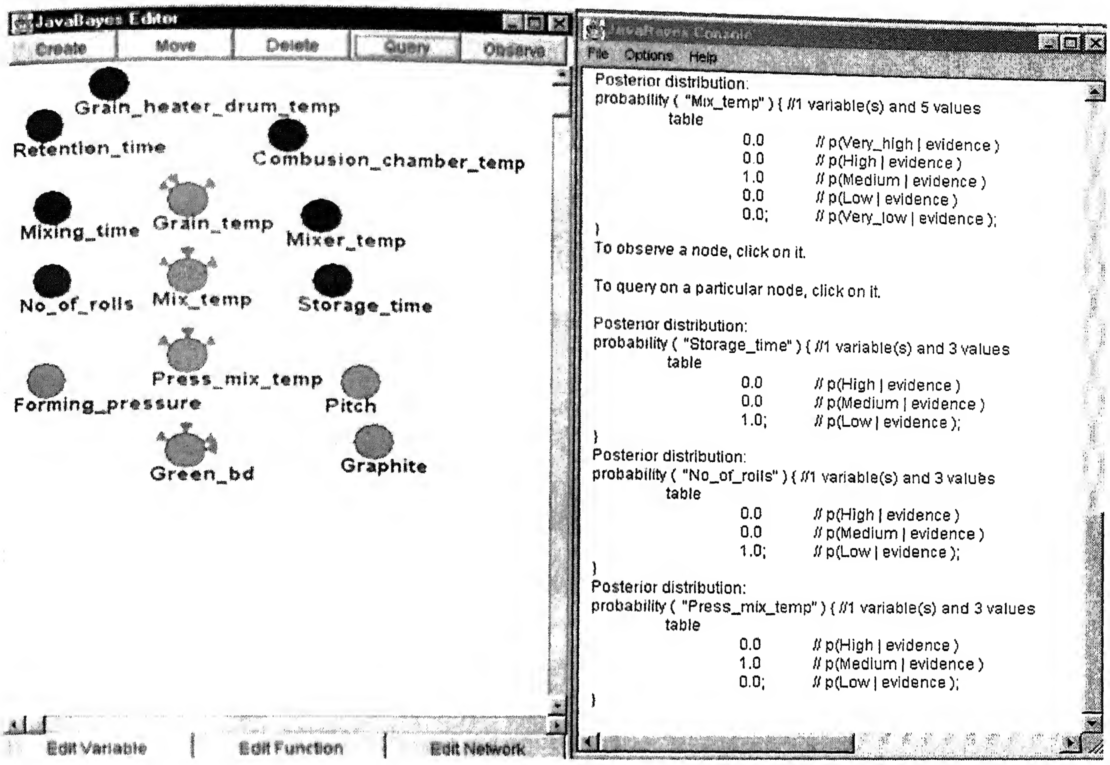


Fig 6.7 Probability Prediction Obtained for PMT (Medium=1)

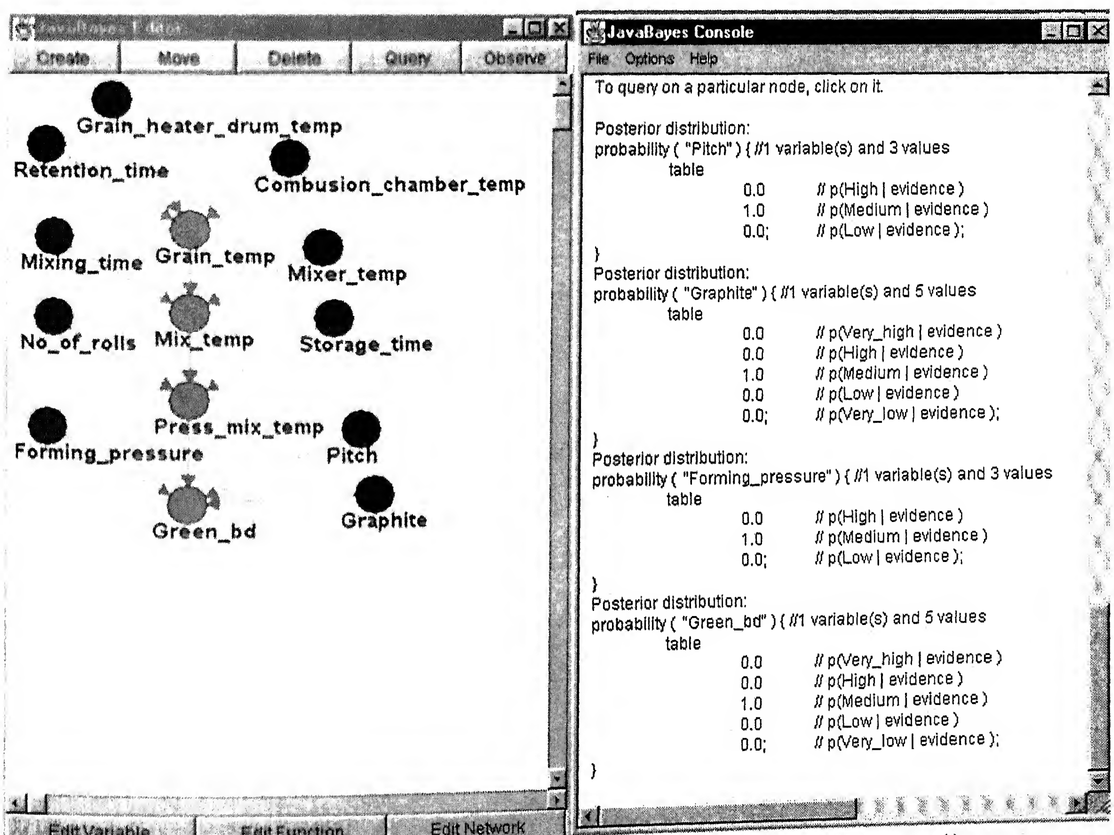


Fig 6.8 Probability Prediction Obtained for GBD (Medium=1)

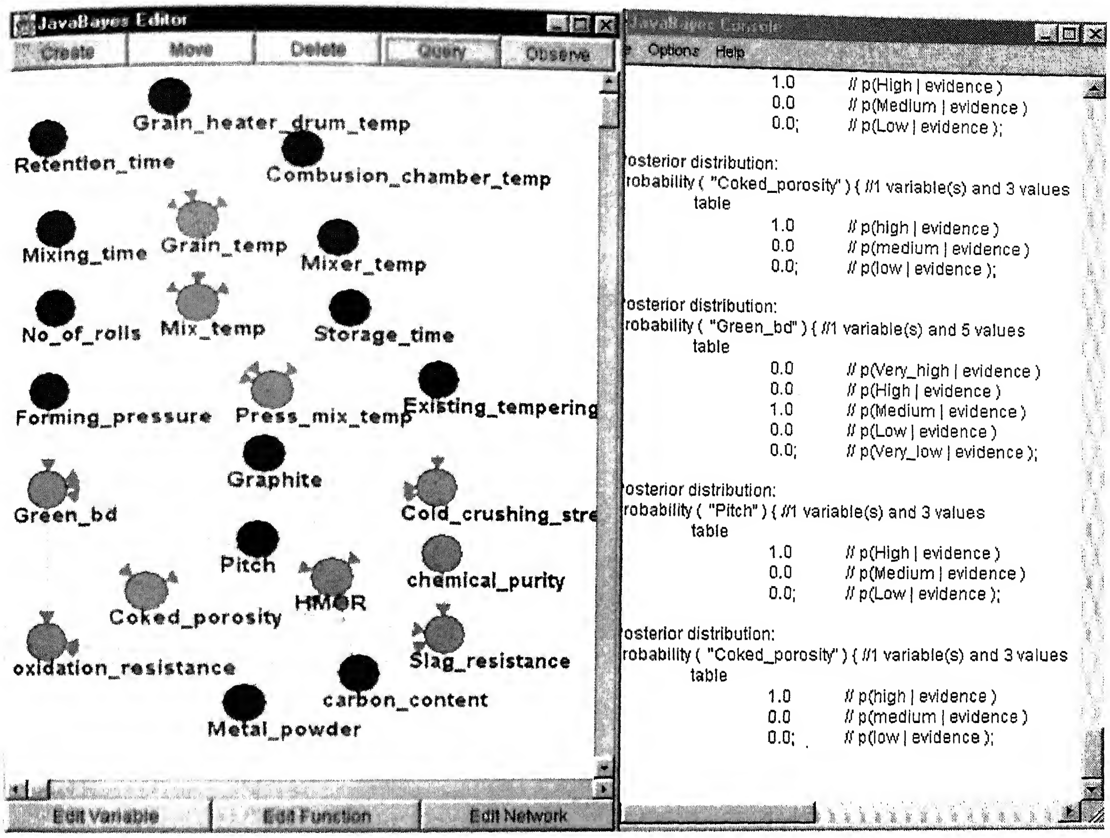


Fig 6.9 Probability Prediction Obtained for Coked Porosity (High=1)

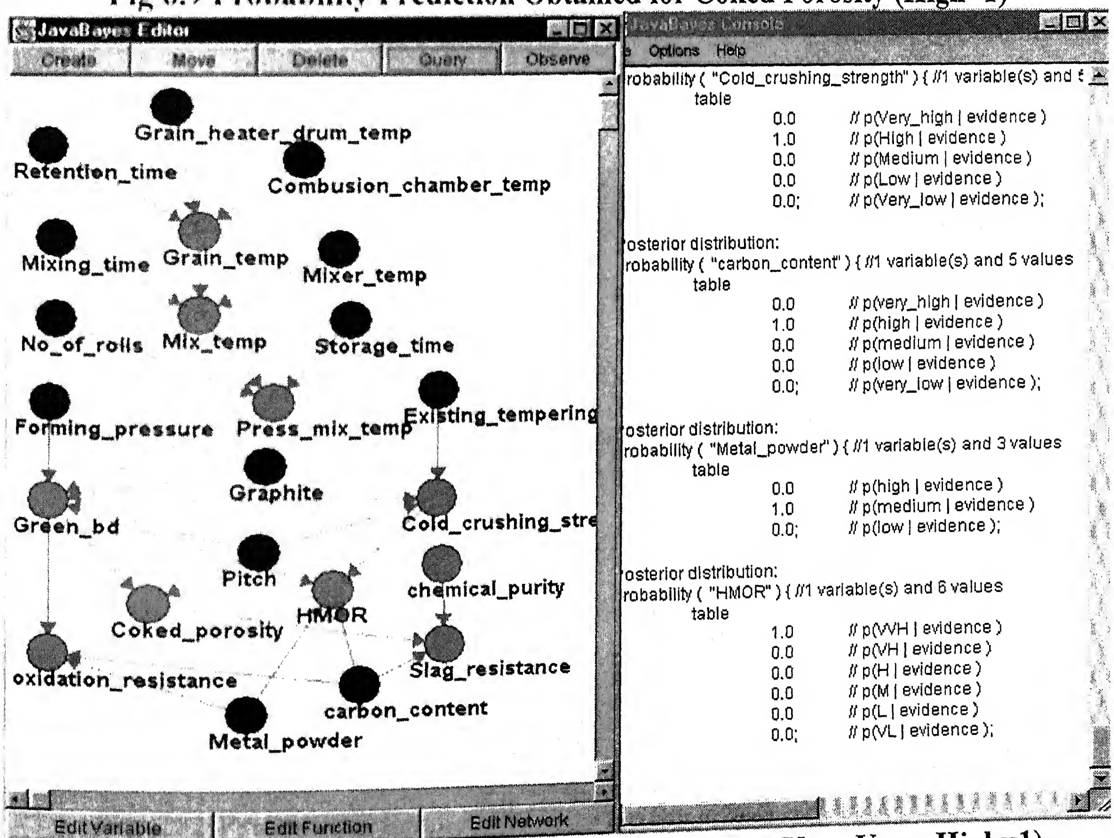


Fig 6.10 Probability Prediction Obtained for HMOR (Very-Very- High=1)

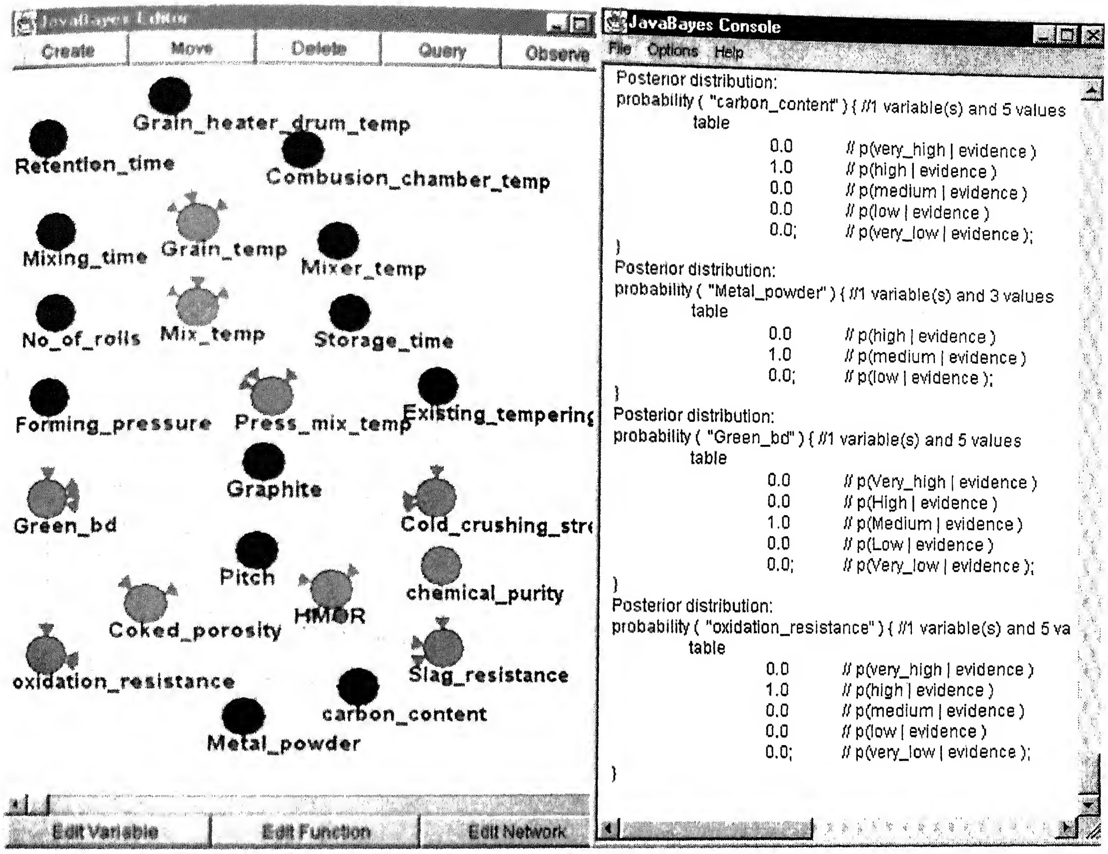


Fig 6.11 Probability Prediction Obtained for Oxidation Resistance (High=1)

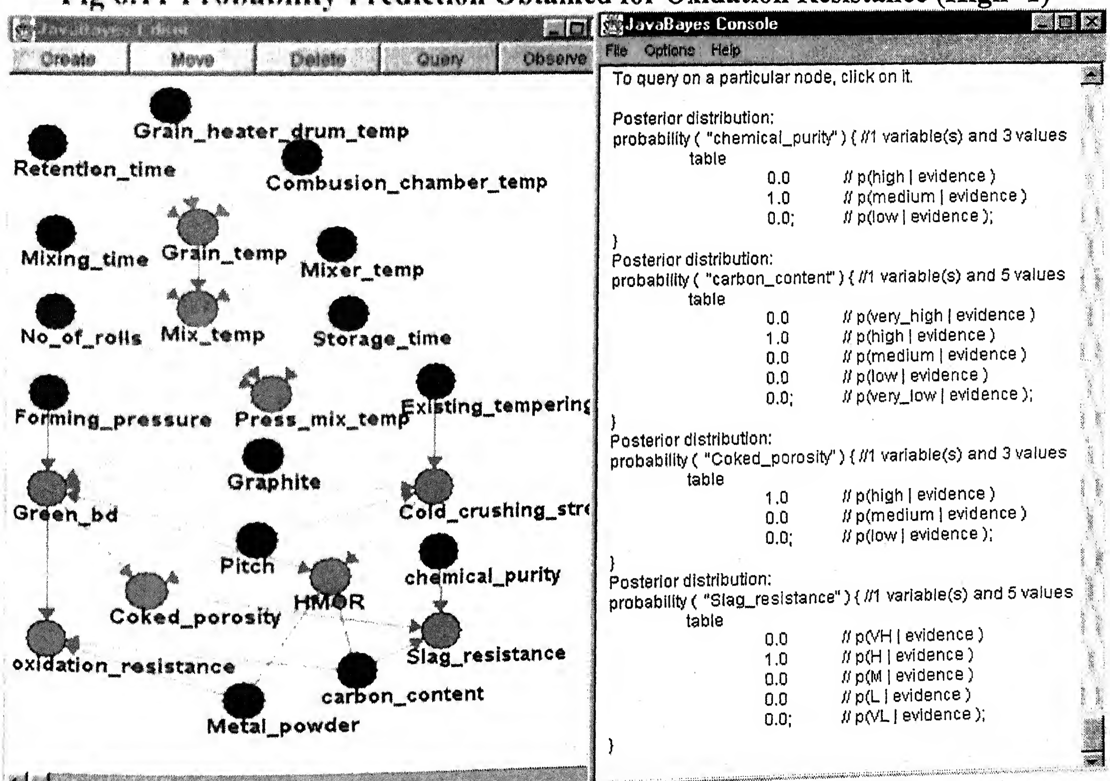


Fig 6.12 Probability Prediction Obtained for Slag Resistance (High=1)

MEMBERSHIP FUNCTIONS FOR ALL ATRIBUTES OF EXPERT BASED SYSTEM

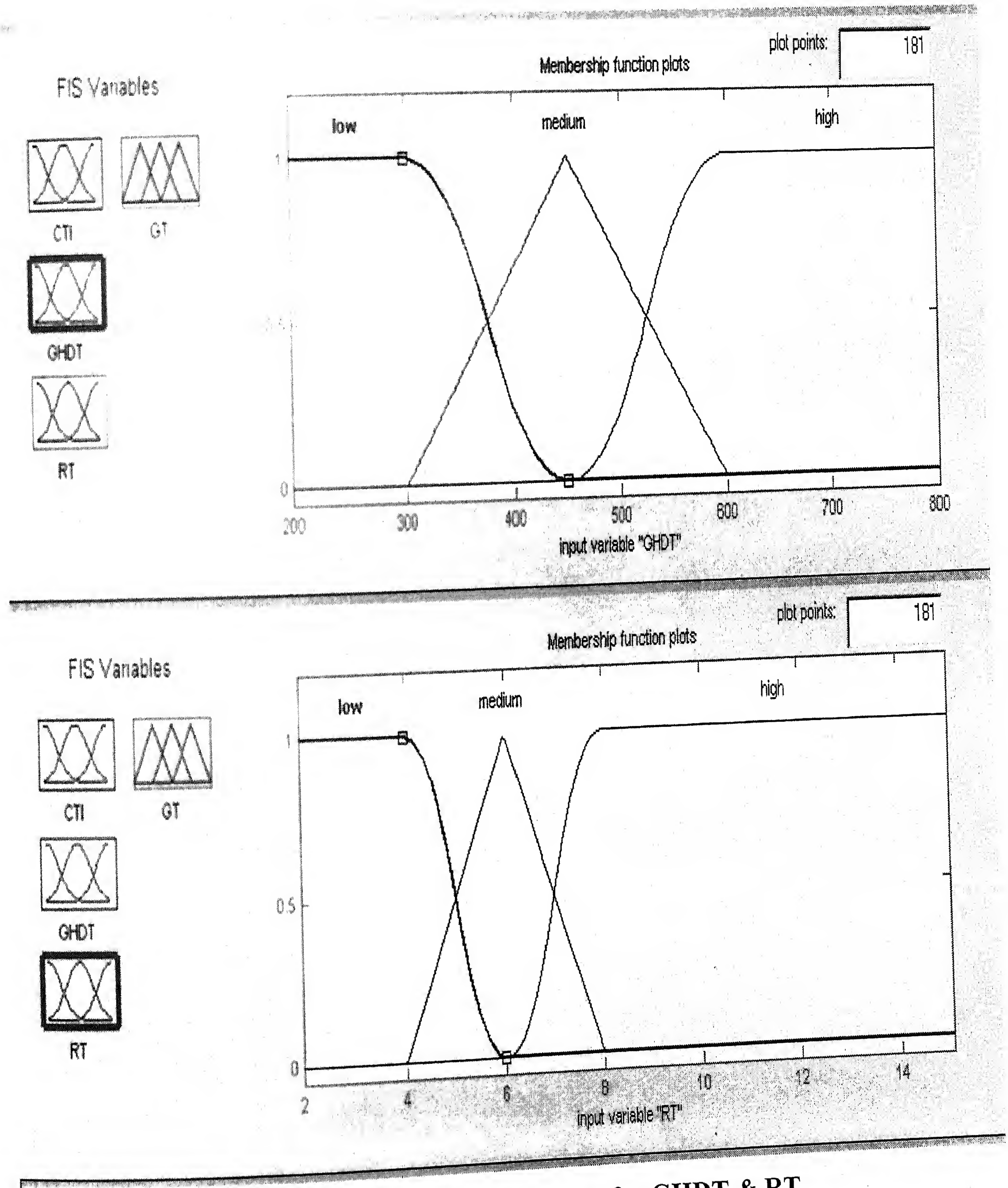


Fig 7.11Memberships functions for GHDT & RT

Mix Temperature Sub system

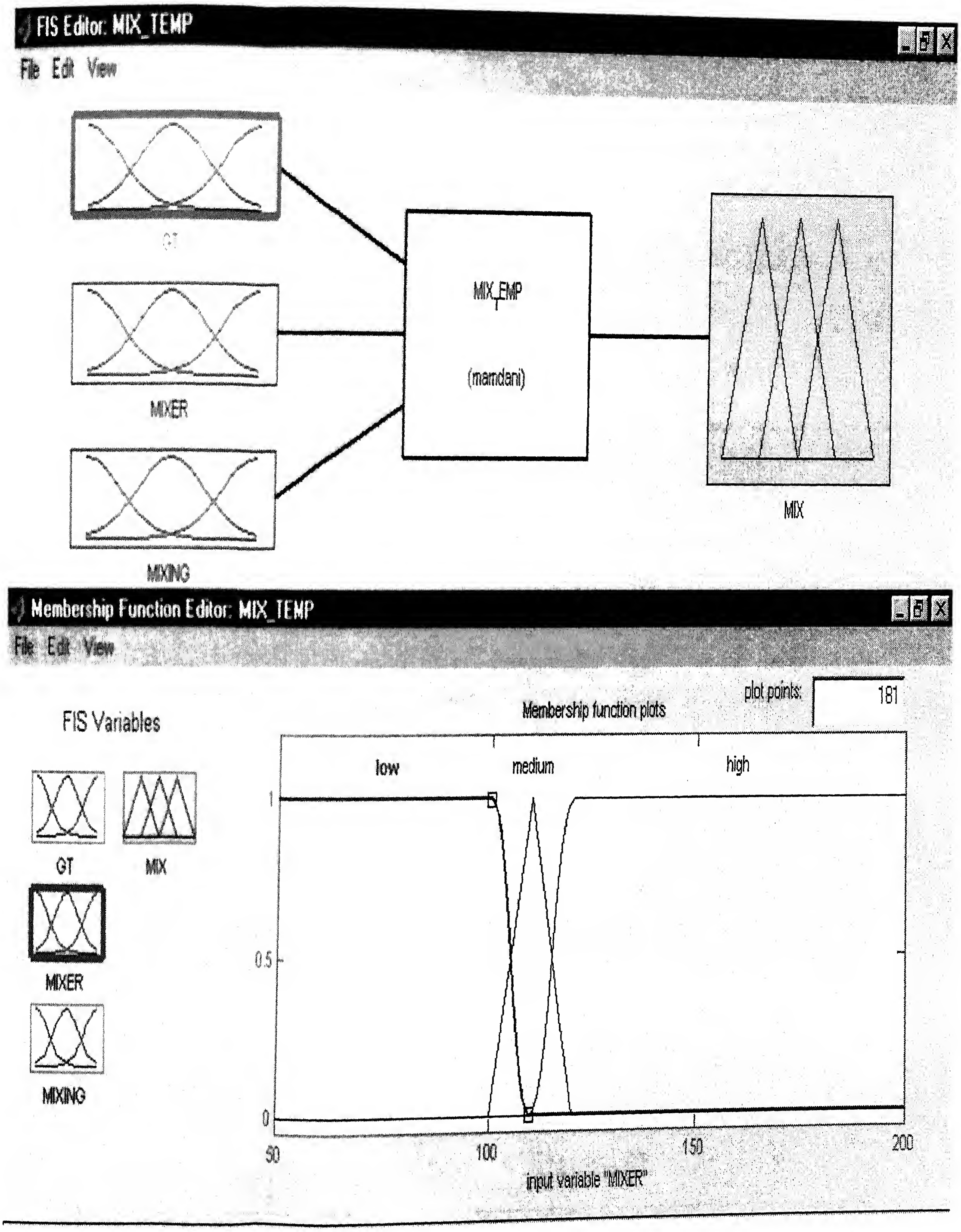


Fig 7.12 FIS structure & membership function for Mixer Temperature

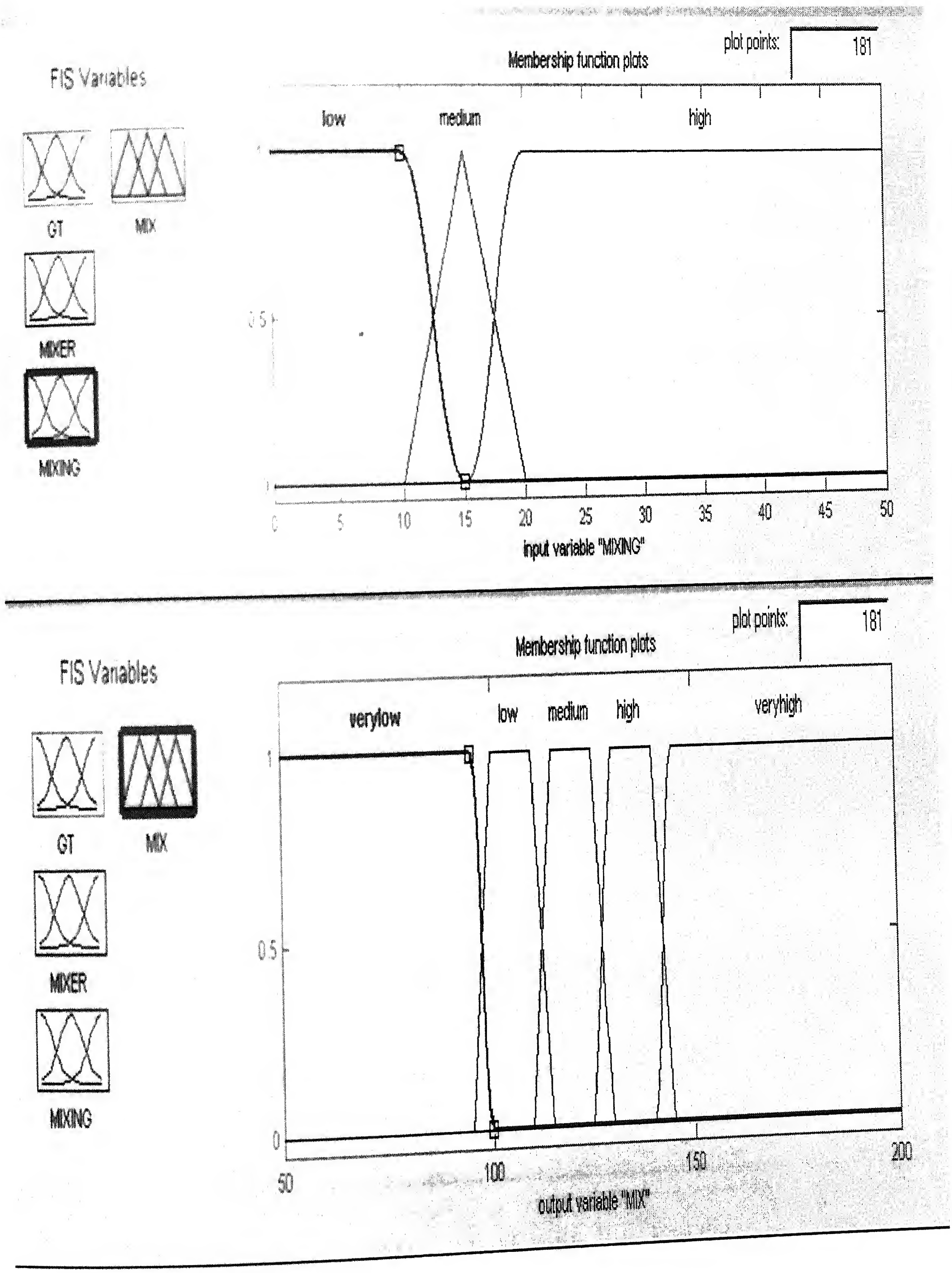


Fig 7.13 Membership function for Mixing Time & Mix Temperature

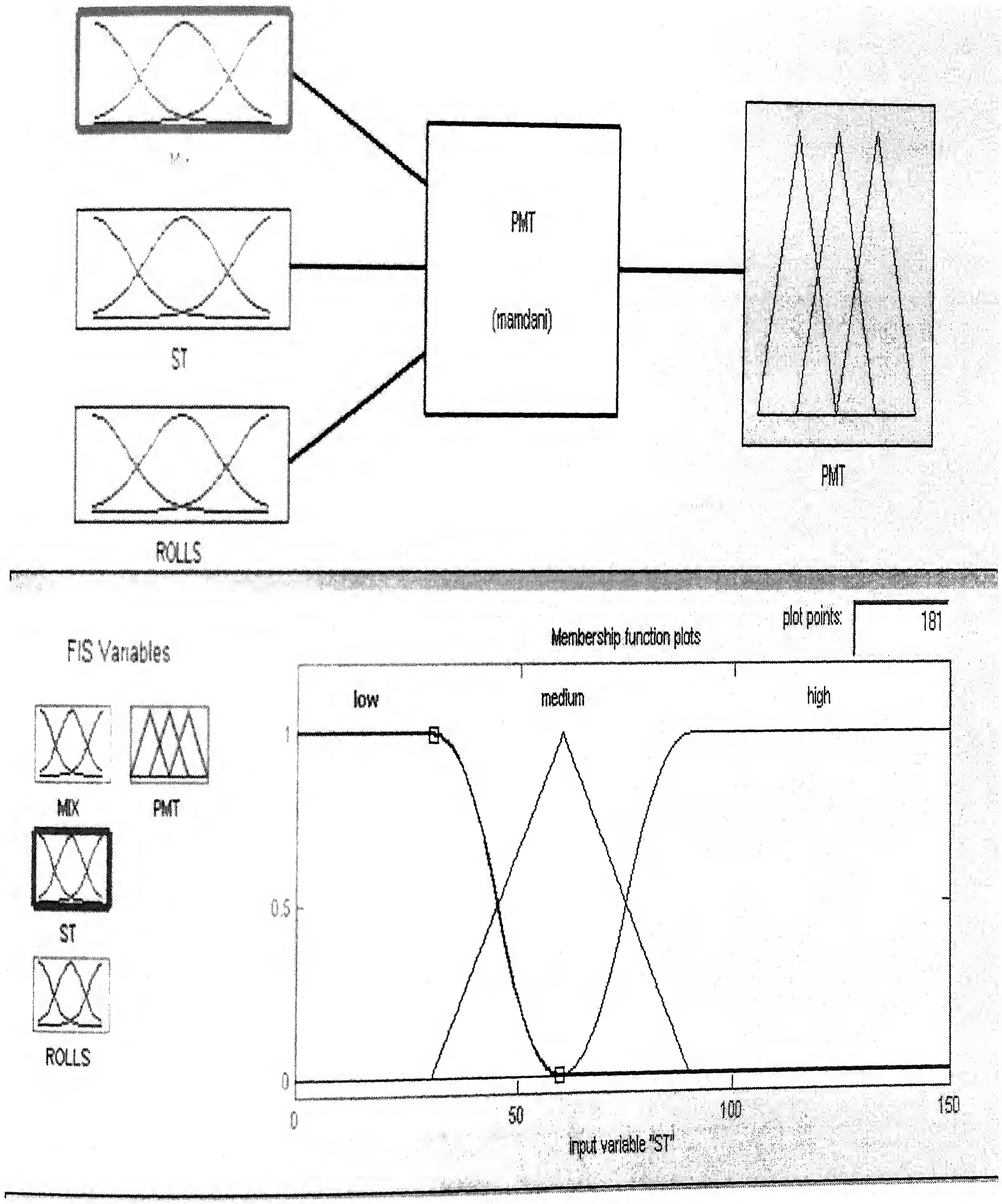


Fig 7.14 Fis Editor for PMT & Membership Function for Storage Time

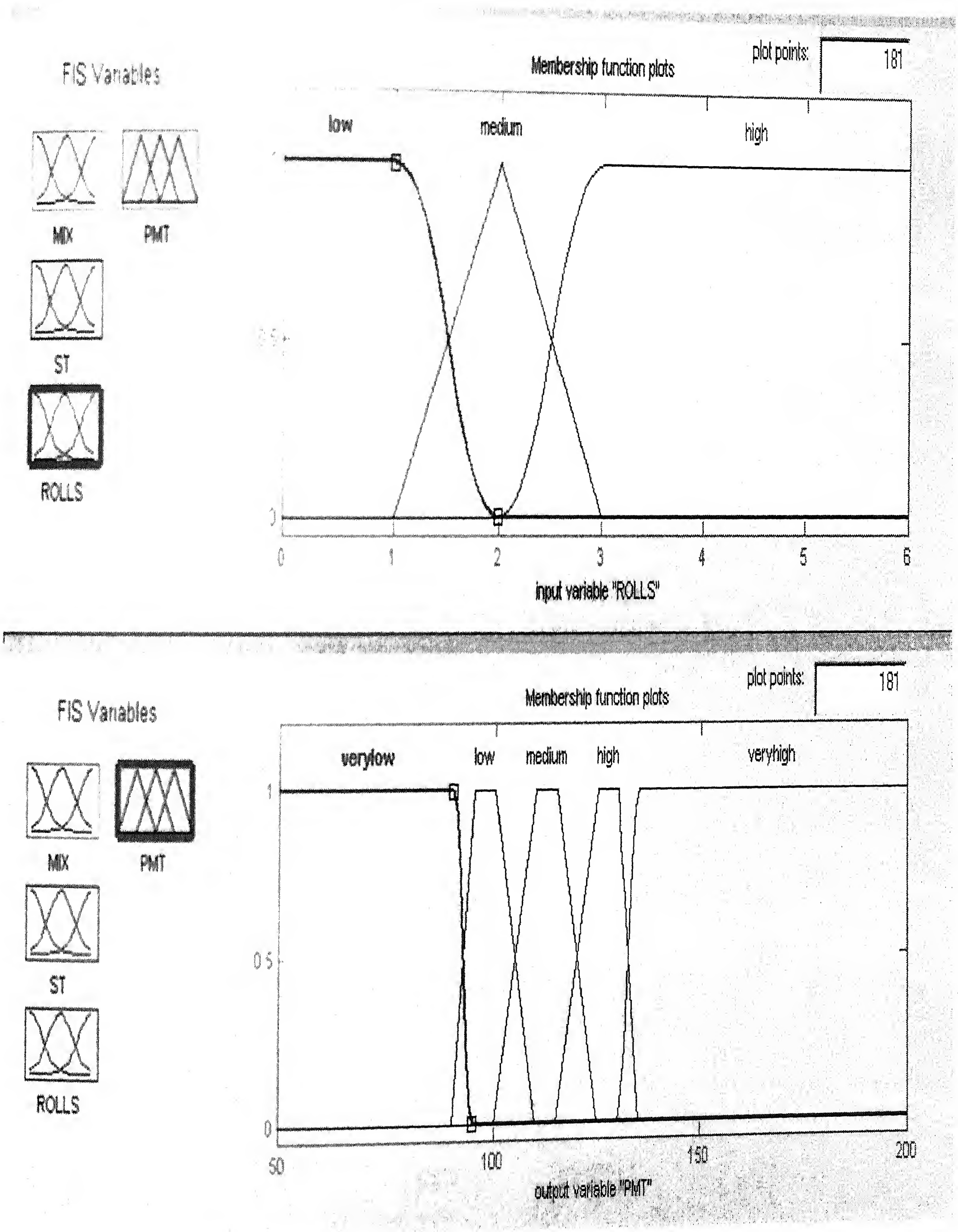


Fig 7.15 Membership Functions for Number of Rolls & PMT

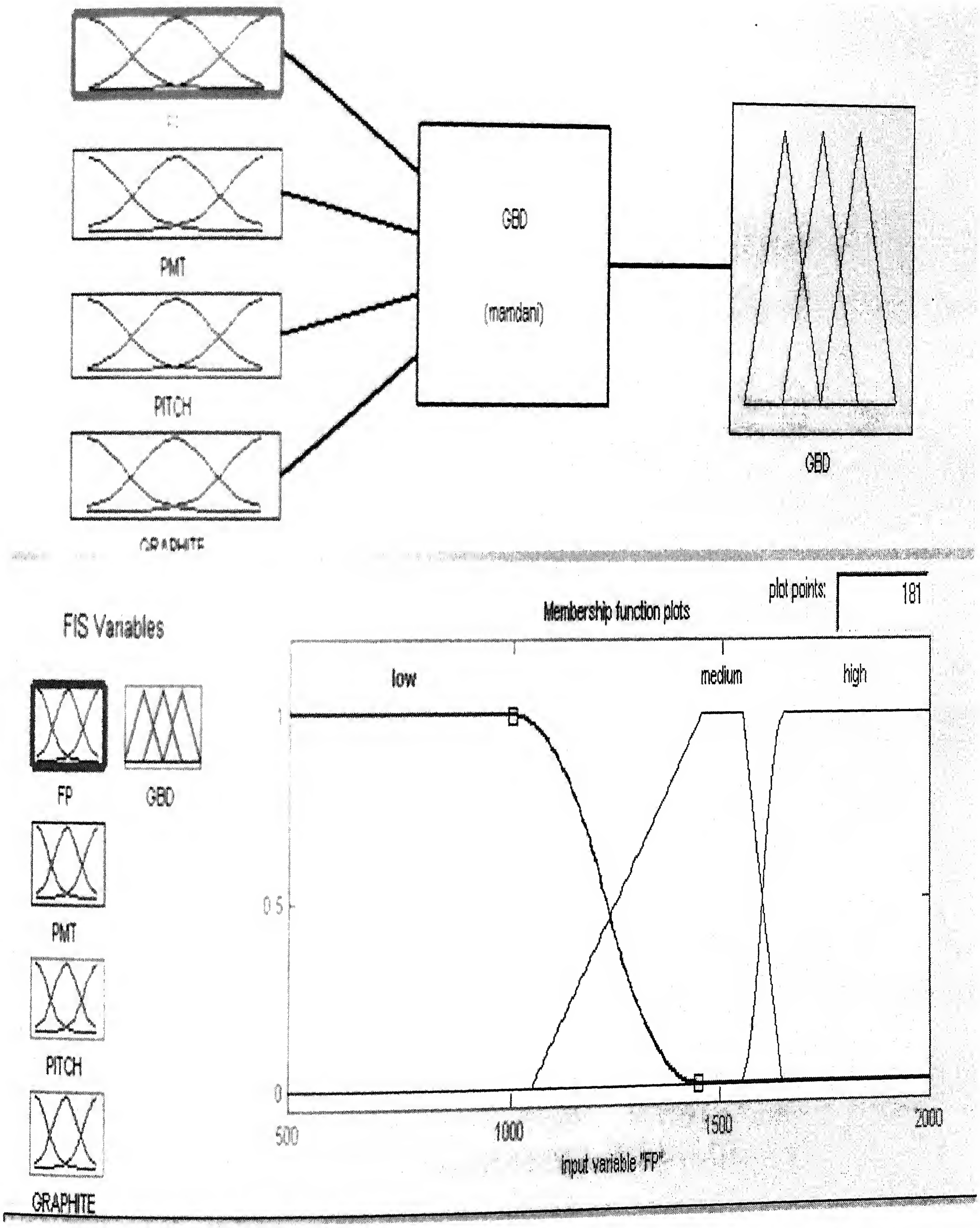


Fig 7.16 Fis Editor for GBD & Membership Function for Forming Pressure

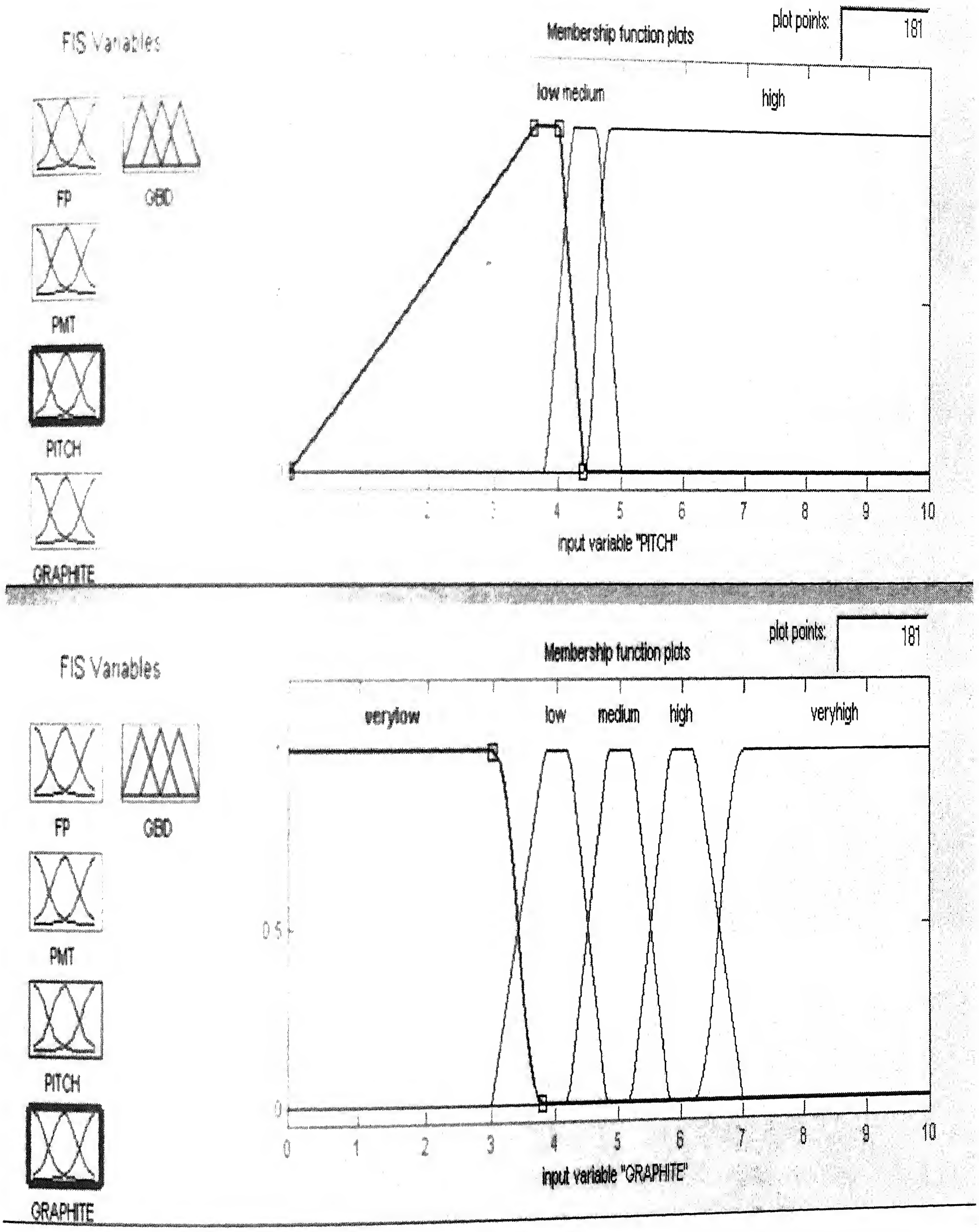


Fig 7.17 Membership Functions for Pitch & Graphite

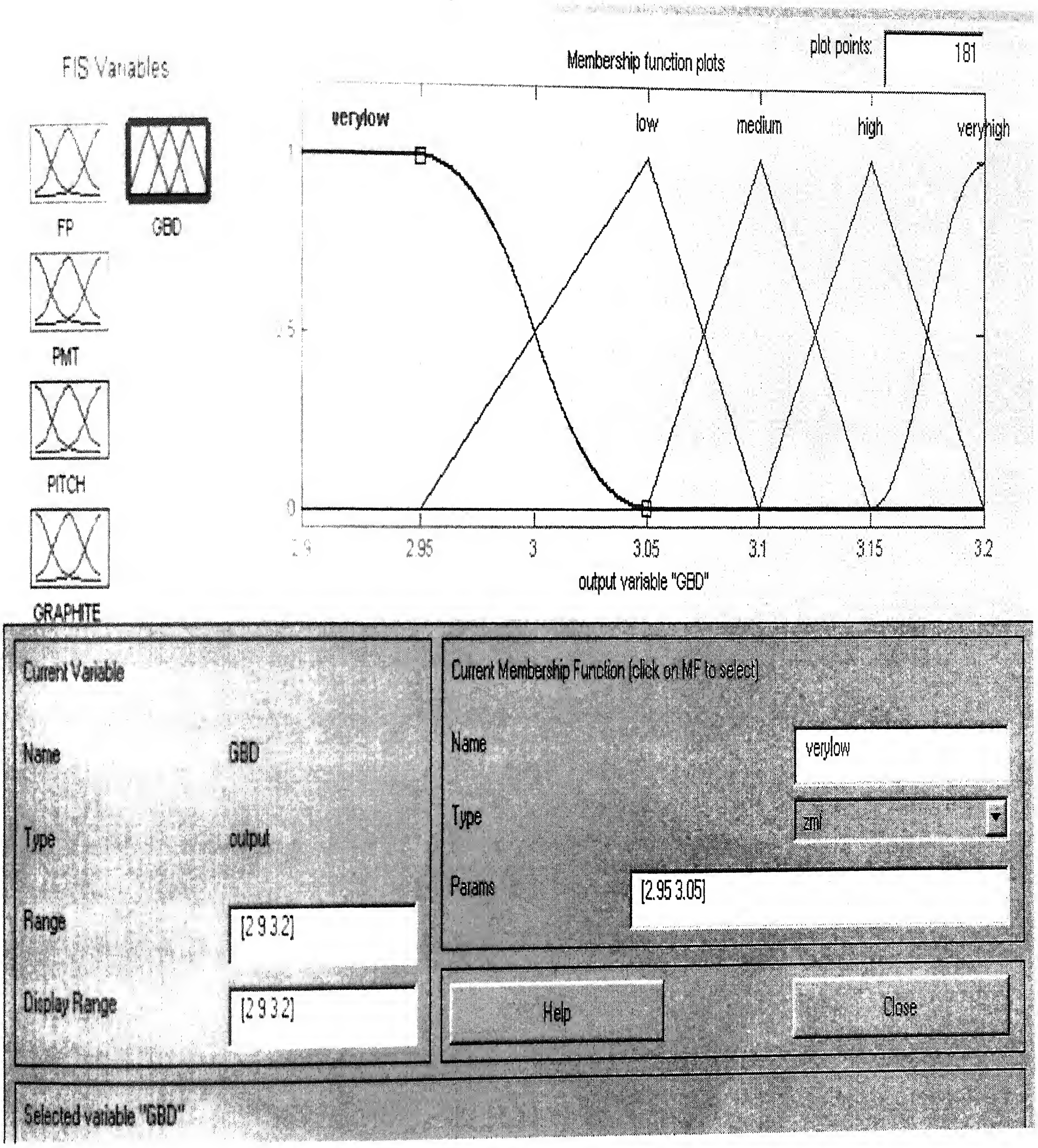


Fig 7.18 Membership Functions for Green Bulk Density

SURFACE PLOTS FOR ALL EXPERT SUB SYSTEMS UPTO GBD

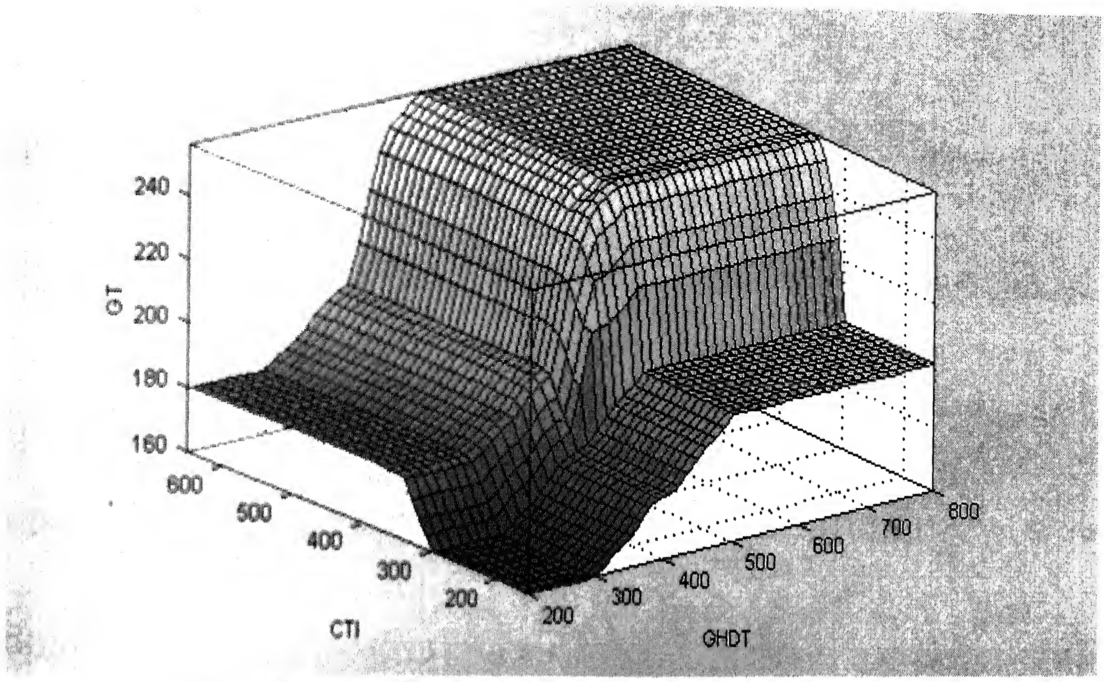


Fig 7.19 Surface Plot for GHDT & CTI Vs Grain Temperature

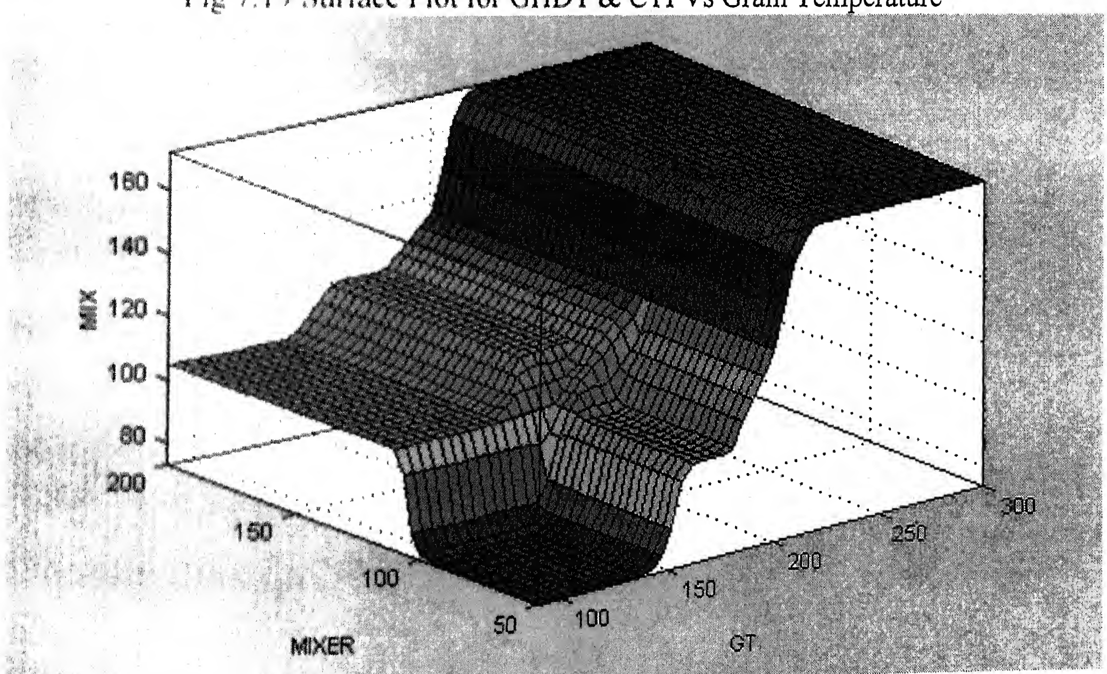


Fig 7.20 Surface Plot for GT & Mixer Temperature Vs Mix Temperature

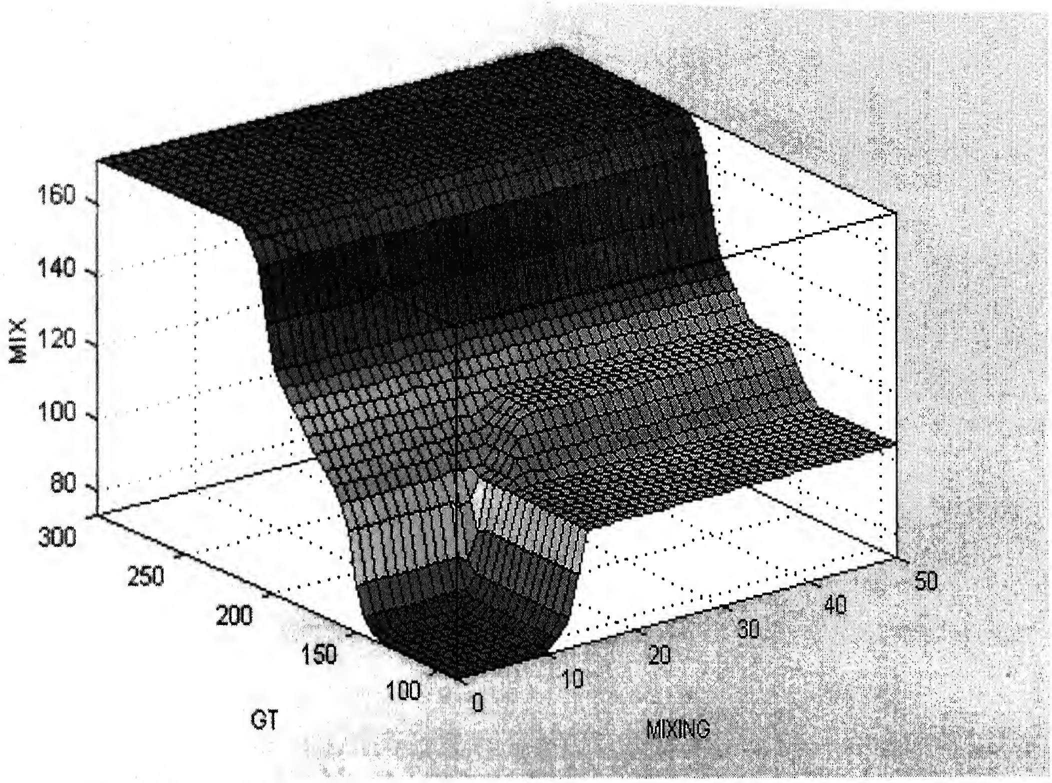


Fig 7.21 Surface Plot for Mixing Time & GT Vs Mix Temperature

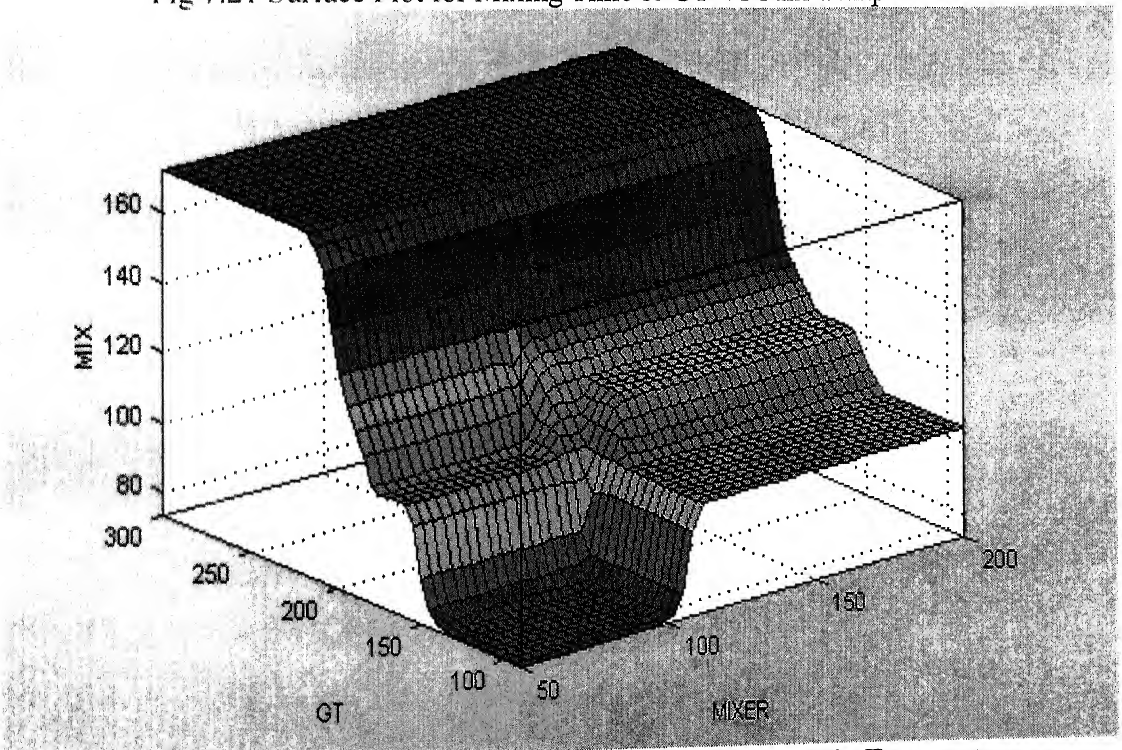


FIG 7.22 Surface Plot for Mixer Temperature & GT Vs Mix Temperature

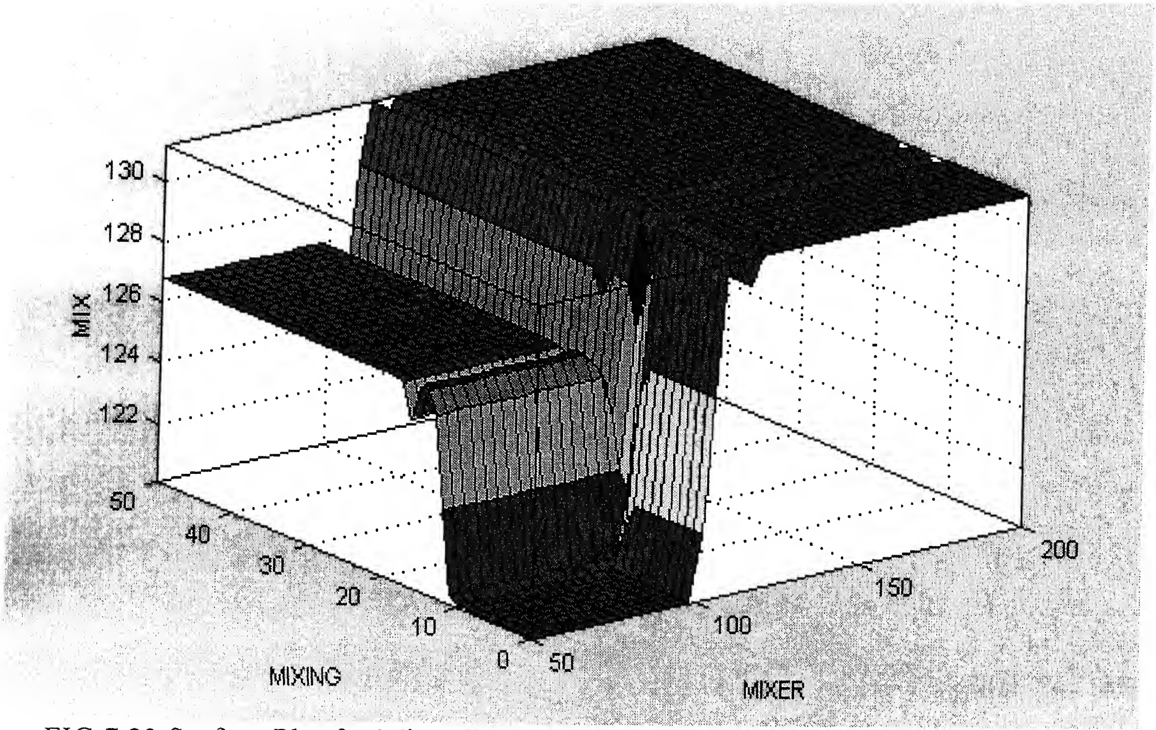


FIG 7.23 Surface Plot for Mixer Temperature & Mixing Time Vs Mix Temperature

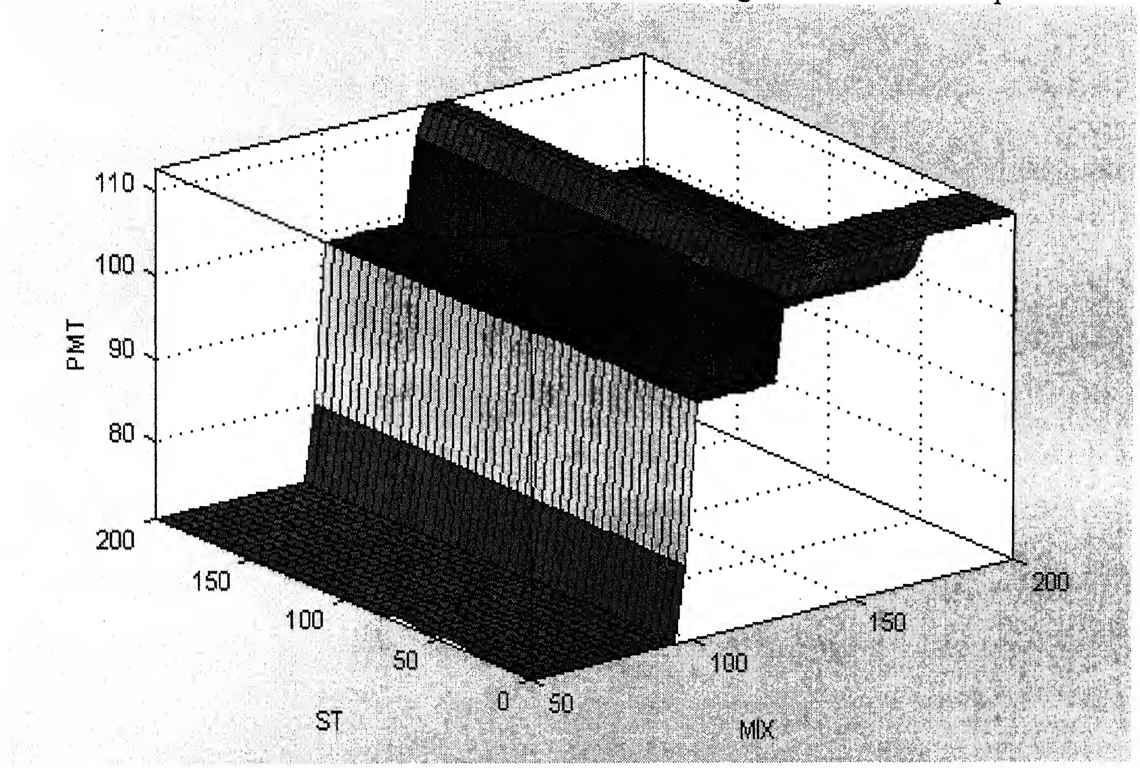


FIG 7.24 Surface Plot for Mix Temperature & Storage Time Vs PMT

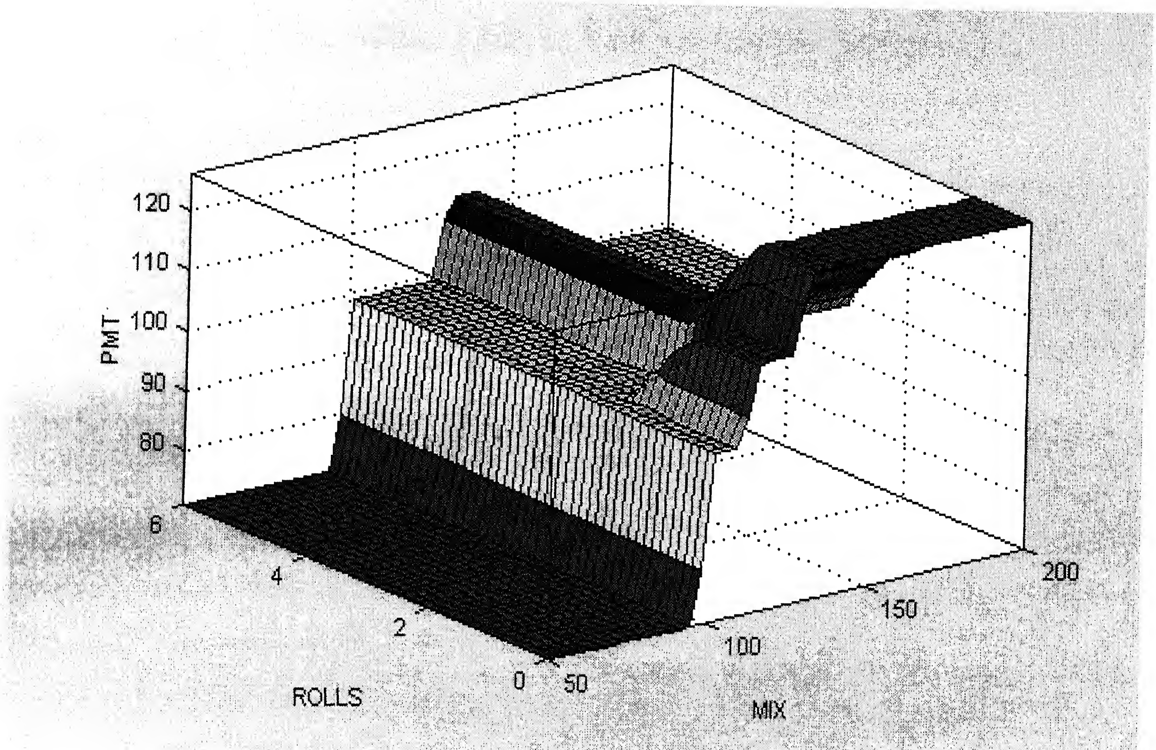


FIG 7.25 Surface Plot for Mix Temperature & Number of Rolls Vs PMT

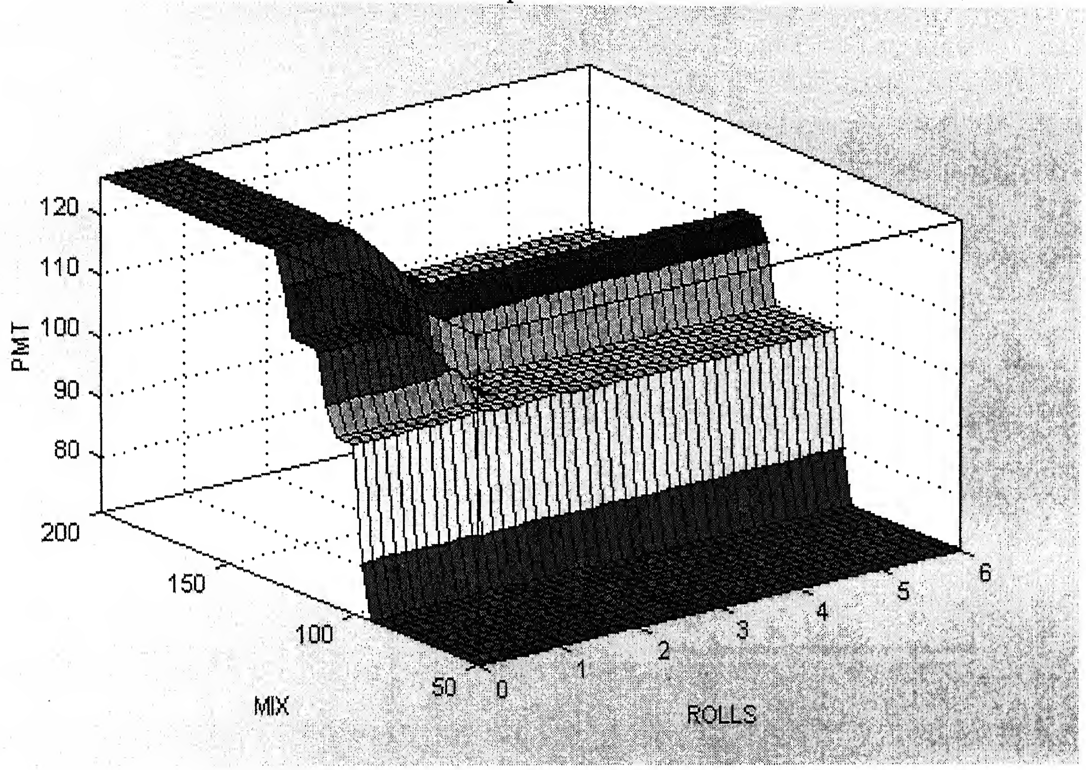


FIG 7.26 Surface Plot for Number of Rolls & Mix Temperature Vs PMT

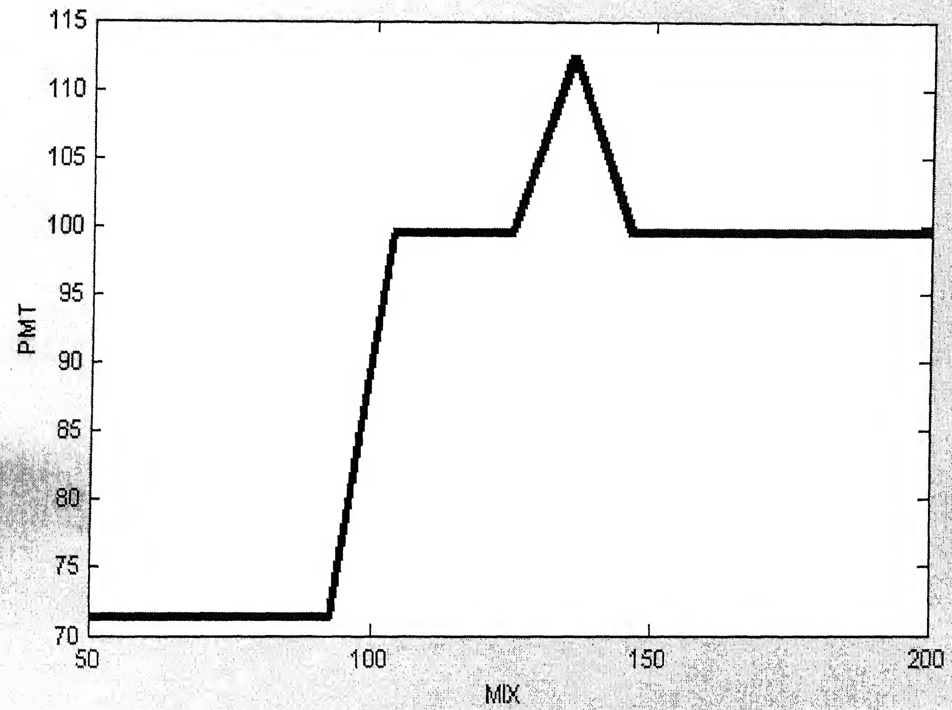


FIG 7.27 Plot for Mix Temperature Vs PMT

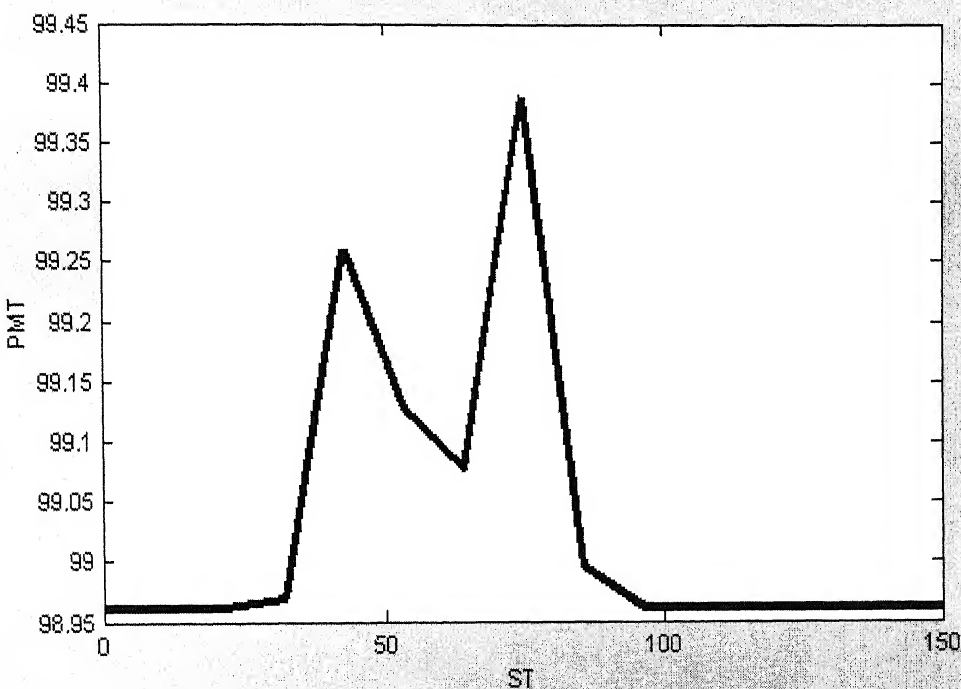


FIG 7.28 Plot for Storage Time Vs PMT

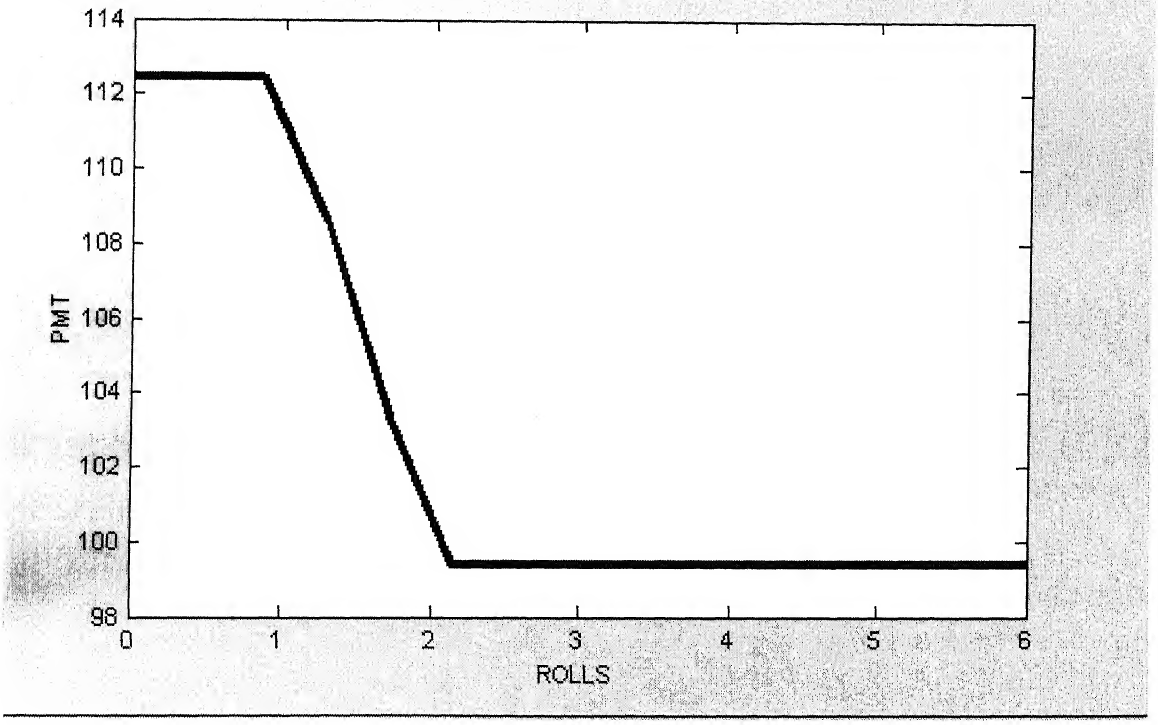


FIG 7.29 Plot for Number of Rolls Vs PMT

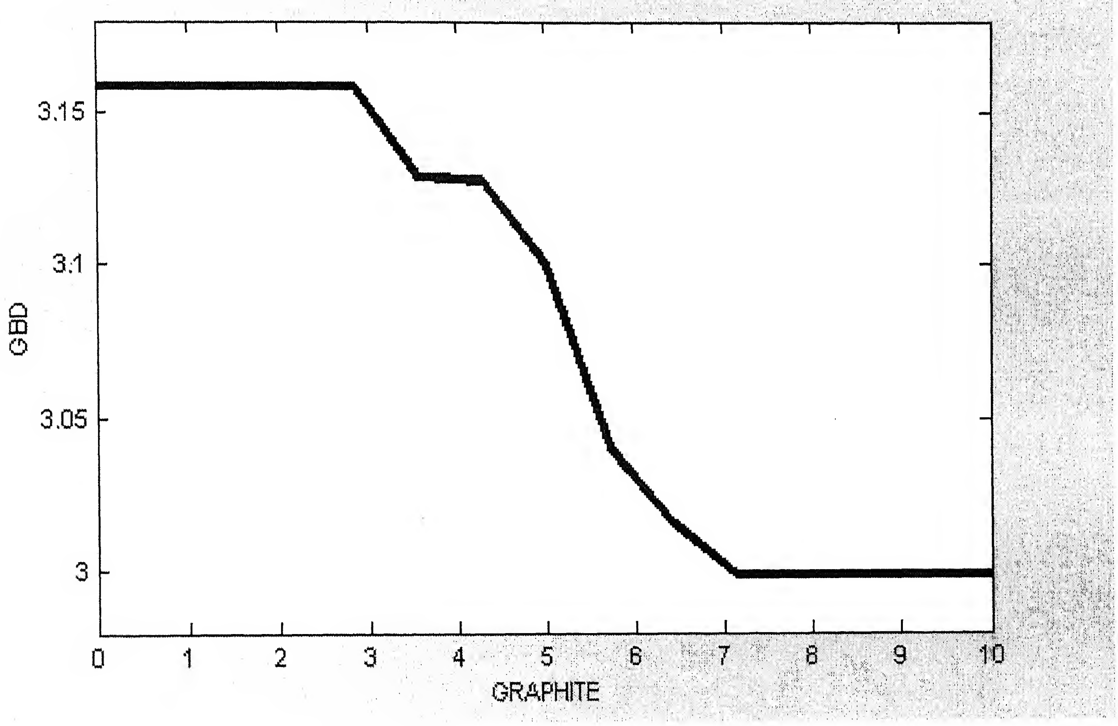


FIG 7.30 Plot for Graphite Vs GBD

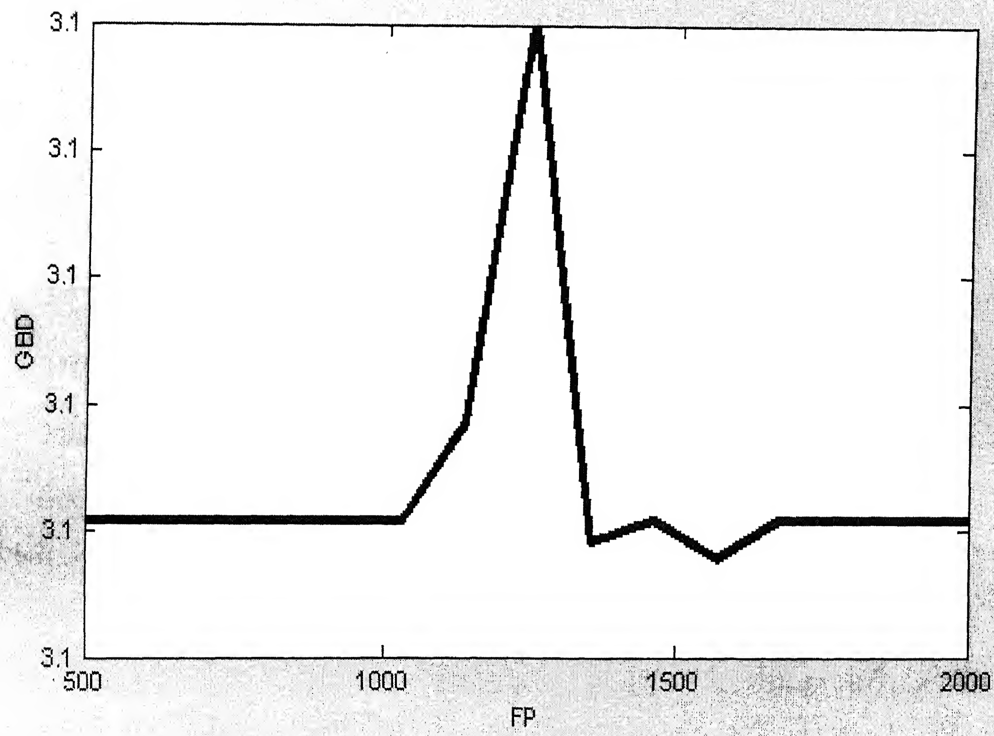


FIG 7.31 Plot for Forming Pressure Vs GBD

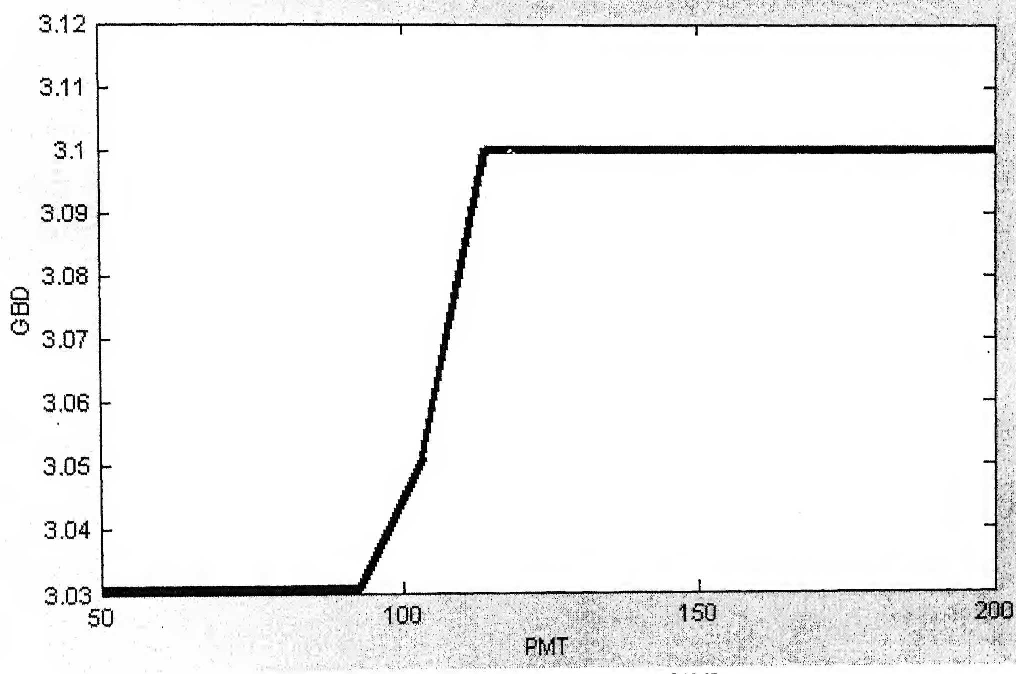


FIG 7.32 Plot for PMT Vs GBD

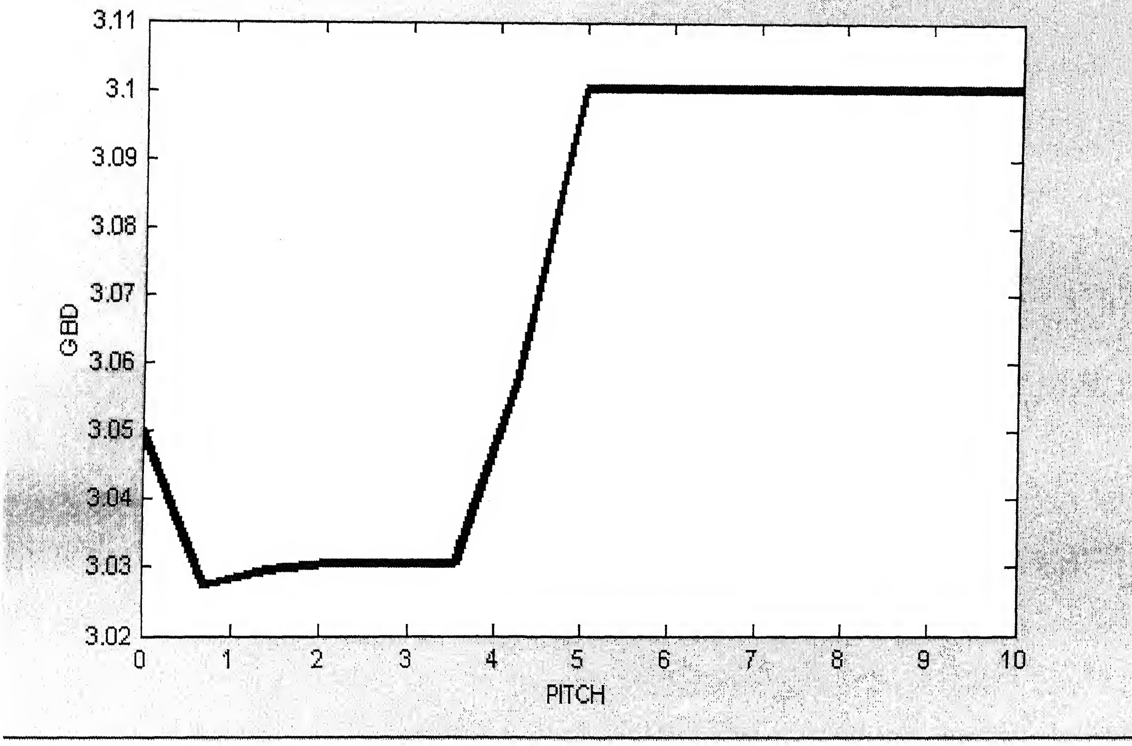


FIG 7.33 Plot for Pitch Vs GBD

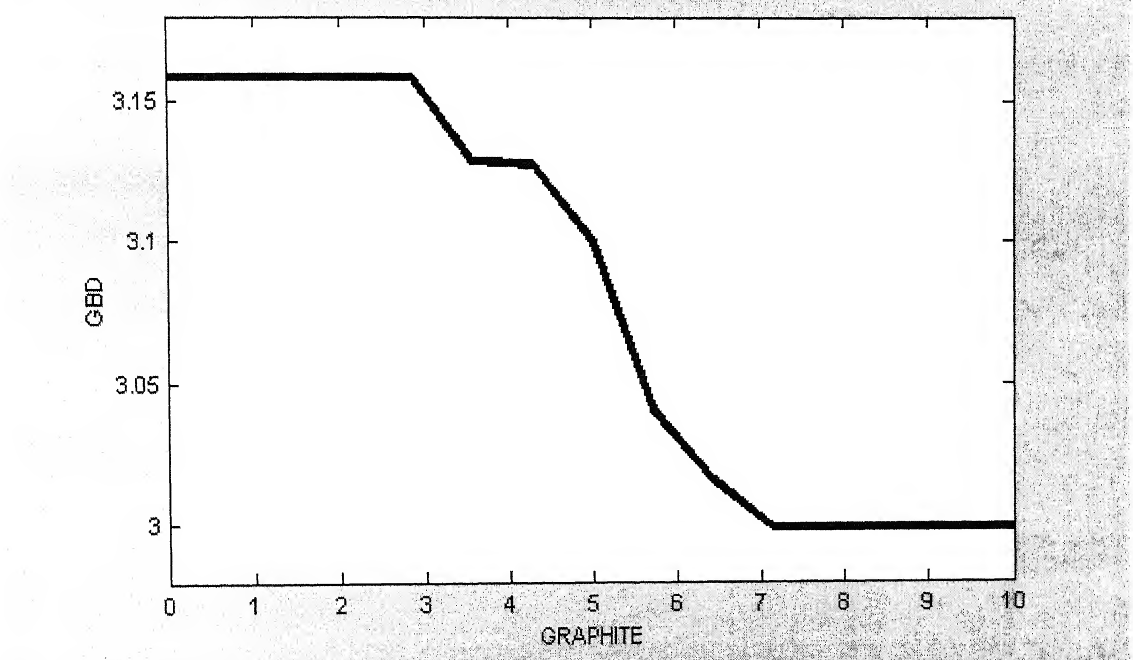
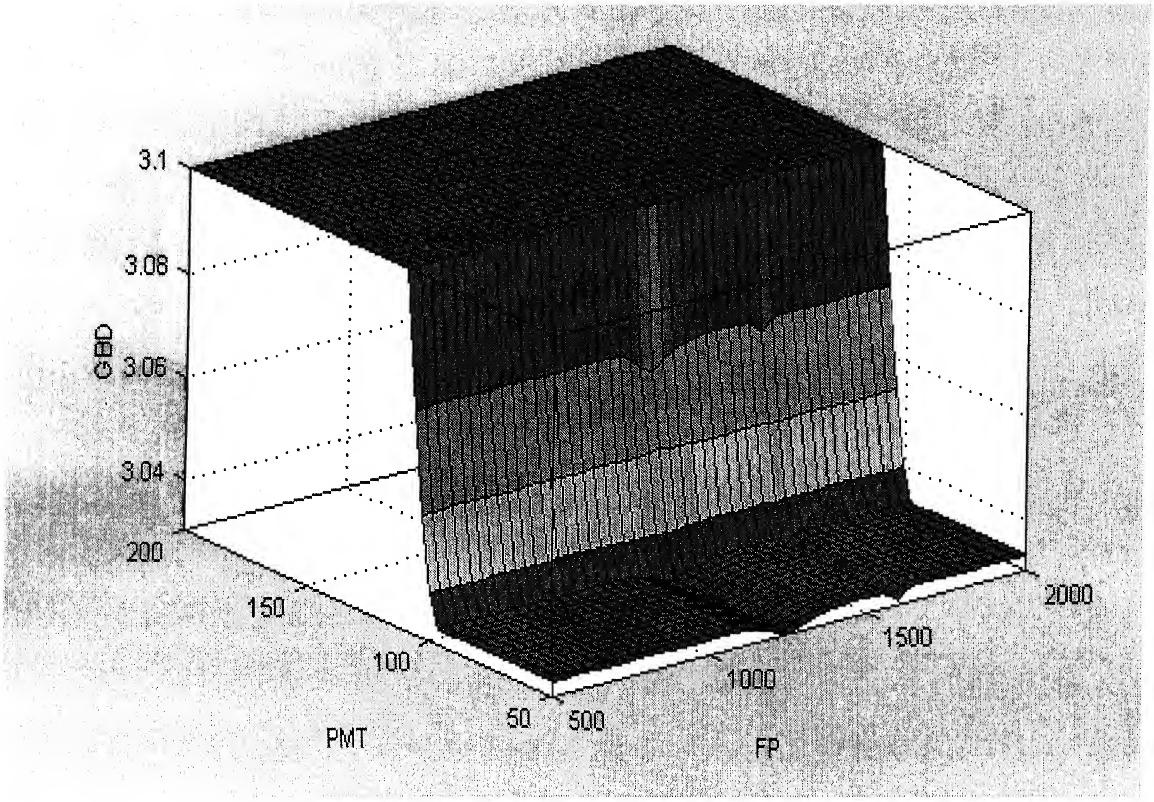
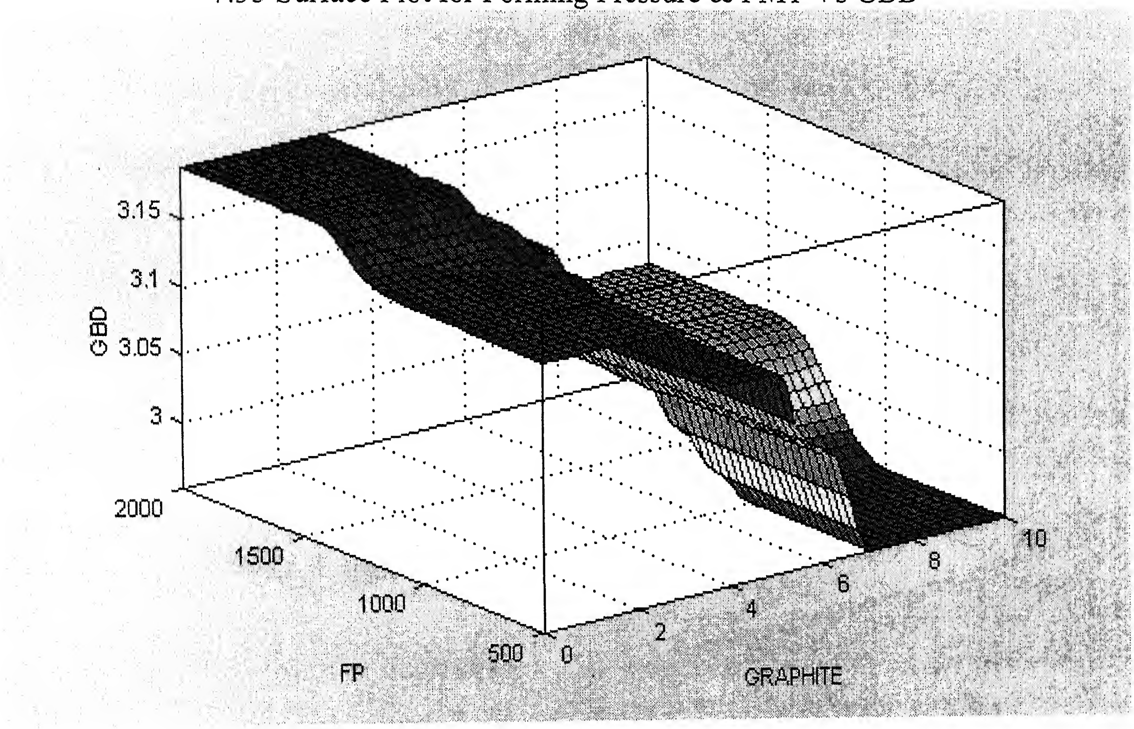


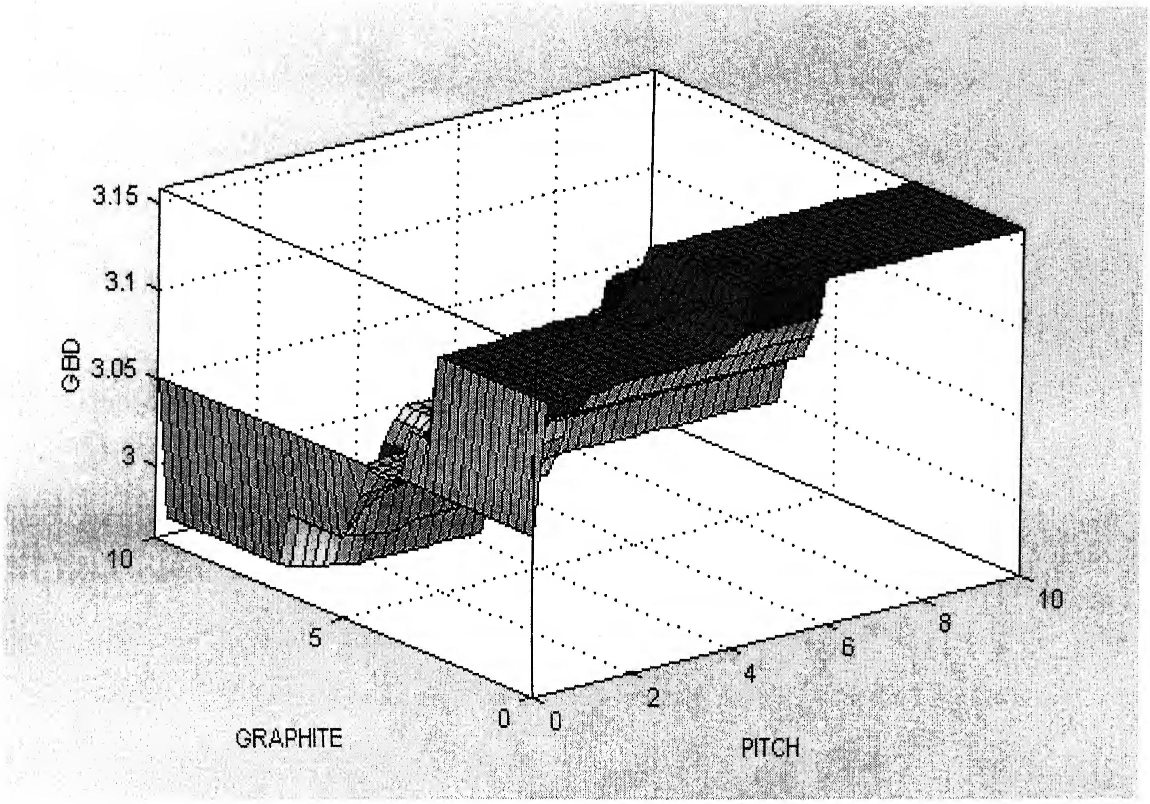
FIG 7.34 Plot for Graphite Vs GBD



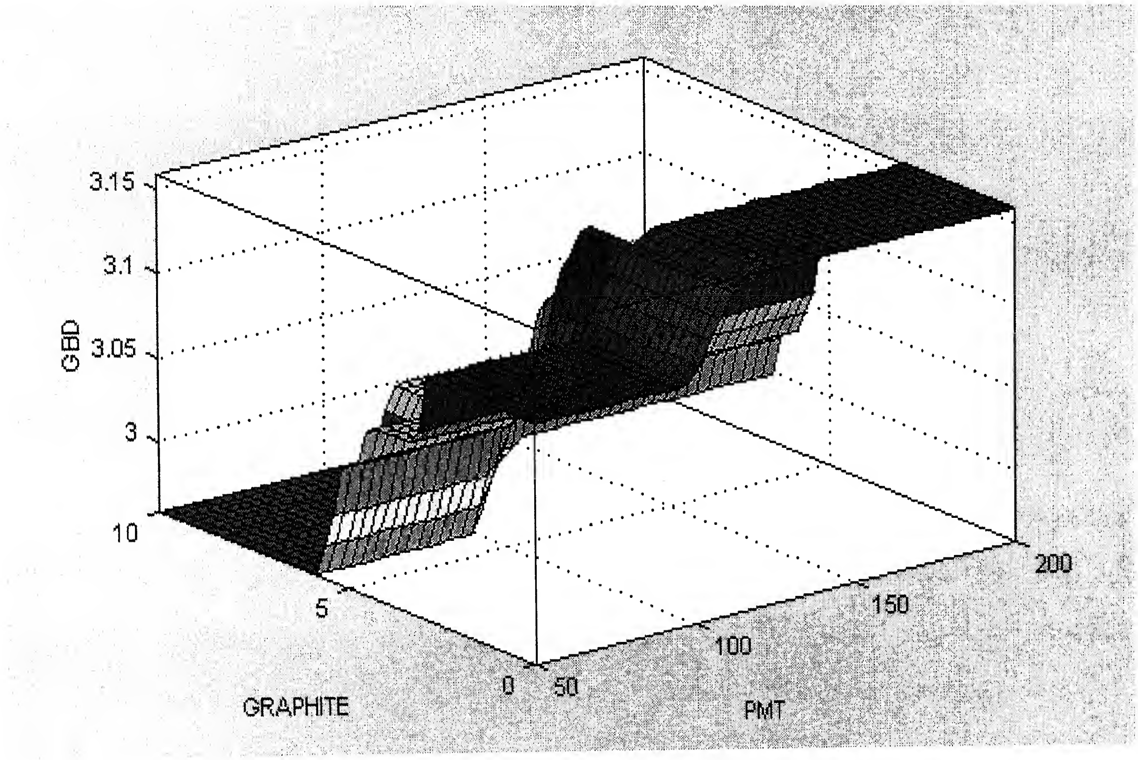
7.35 Surface Plot for Forming Pressure & PMT Vs GBD



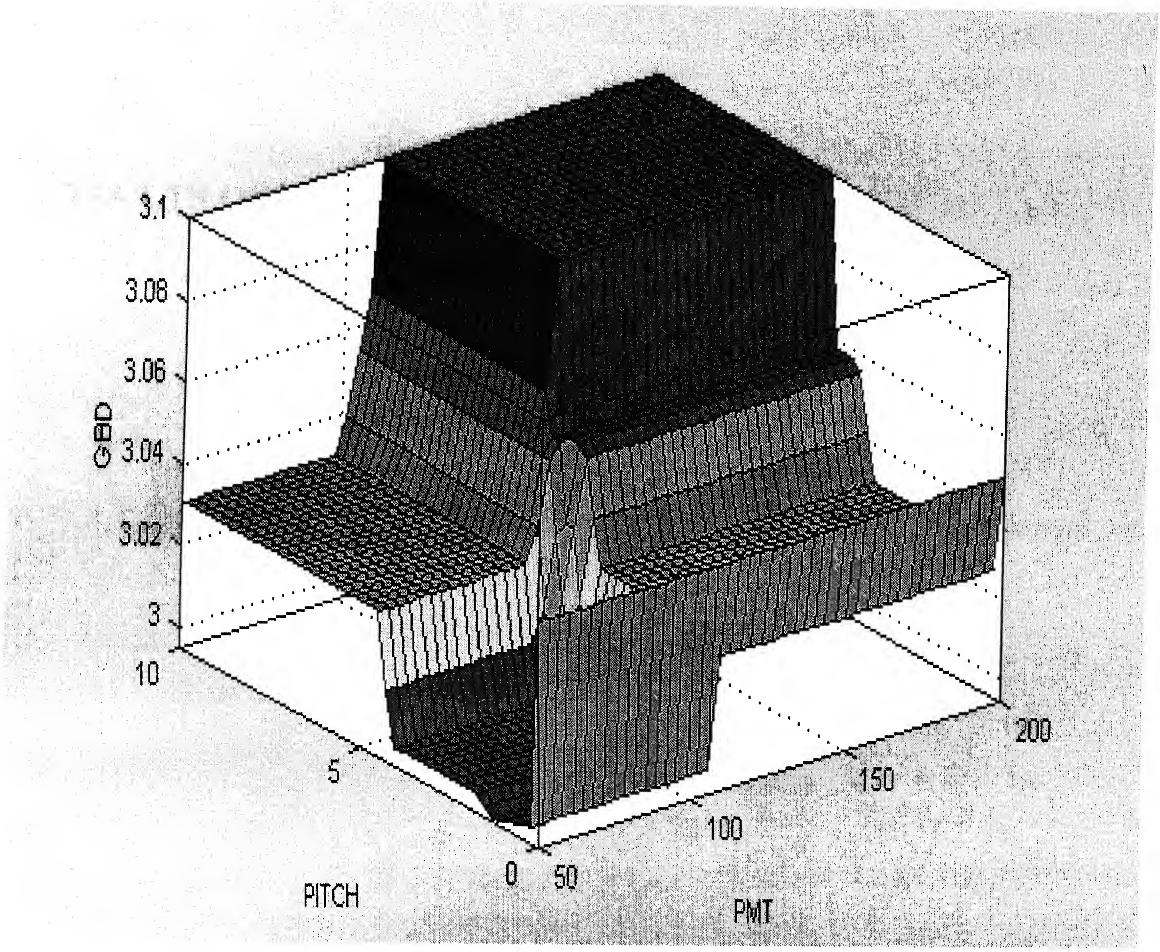
7.36 Surface Plot for Graphite & FP Vs GBD



7.37 Surface Plot for Pitch & Graphite Vs GBD



7.38 Surface Plot for PMT & Graphite Vs GBD



7.39 Surface Plot for PMT & Pitch Vs GBD

RESULTS OBTAINED ON SAMPLE DATA APPLIED TO THE FIS SYSTEMS

Table 7.11 Results for Grain Temperature

INPUT			DESIRED OUTPUT RANGE	OUTPUT
CTI	GHDT	RT	GRAIN TEMP	GRAIN TEMP
376	450	5.51	VERY HIGH	233.124977
378	456	4.6	HIGH	183.582759
380	461	4	MEDIUM	172.120962
385	429	5.6	HIGH	215.598978
391	433	4.8	MEDIUM	183.15543
398	446	4.1	MEDIUM	181.862239
400	318	5.7	MEDIUM	177.91045
379	321	5	LOW	165.388546
383	326	4.2	VERY LOW	130.761992
331	463	5.8	VERY HIGH	240.914753
336	466	5.2	HIGH	206.703371
342	470	4.3	MEDIUM	180.014468
349	367	5.9	HIGH	190.802298
350	374	5.4	MEDIUM	183.601613
358	376	4.4	LOW	158.050082
363	309	6	MEDIUM	182.570329
368	312	5.1	LOW	164.364449
372	315	4.49	VERY LOW	135.220846
309	472	5.51	HIGH	215.998286
311	475	4.6	MEDIUM	178.196806
315	479	4	LOW	166.387609
320	358	5.6	MEDIUM	156.267735
320	363	4.8	MEDIUM	150.800699
321	367	4.1	LOW	145.29914
319	300	5.7	LOW	147.810478
313	304	5	VERY LOW	125.456792
301	306	4.2	VERY LOW	122.39157

Table 7.12 Results for Mix Temperature

INPUT			DESIRED OUTPUT RANGE	OUTPUT
GRAIN TEMP	MIXER TEMP	MIXING TIME	MIX TEMP	MIX TEMP
211	115.1	17.51	VERY HIGH	164.1858
214	116	12.8	VERY HIGH	166.5776
216	117	10	VERY HIGH	169.1936
218	106	18	VERY HIGH	169.7242
213	108	13	VERY HIGH	164.9027
214	109	10.5	VERY HIGH	167.8098
216	100	18.5	VERY HIGH	169.1936
218	101	14	VERY HIGH	168.9946
212	102	11	VERY HIGH	162.1965
191	118	19	HIGH	128.25
194	119	15	HIGH	130.3551
198	120	11.5	HIGH	133.2981
200	112	19.5	HIGH	135.0412
202	114	16	HIGH	138.7427
204	111	12	HIGH	145.7767
207	103	20	HIGH	156.0311
209	104	17	HIGH	161.5151
201	103	12.4	MEDIUM	127.9053
172	115.1	17.51	MEDIUM	113.378
175	116	12.8	MEDIUM	115.3174
177	117	10	MEDIUM	117.5576
179	106	18	MEDIUM	114.68
183	108	13	MEDIUM	121.3819
186	109	10.5	MEDIUM	124.3532
188	100	18.5	LOW	117.4741
189	101	14	LOW	118.8659
180	102	11	MEDIUM	117.0909
151	118	19	MEDIUM	113.6196
153	119	15	LOW	105.0448
156	120	11.5	LOW	97.69126
159	112	19.5	LOW	106.5645
160	114	16	LOW	106.8816
162	111	12	LOW	107.1373
165	103	20	LOW	109.0165
169	104	17	LOW	111.4782
163	103	12.4	LOW	108.3966

GRAIN TEMP	MIXER TEMP	MIXING TIME	MIX TEMP	MIX TEMP
140	115.1	17.51	LOW	105.0582
141	116	12.8	LOW	83.87308
143	117	10	VERY LOW	75.43555
144	106	18	LOW	85.73656
146	108	13	LOW	85.73656
147	109	10.5	VERY LOW	77.81377
148	100	18.5	VERY LOW	78.81074
149	101	14	VERY LOW	79.97941
142	102	11	VERY LOW	75.63413

Table 7.13 Results for PMT

INPUT			DESIRED OUTPUT RANGE	OUTPUT
MIX	ST	ROLL	PMT	PMT
142.6	76	3	LOW	106.5256
143	79	2	MEDIUM	112.4684
144	82	0	HIGH	125.9054
145	46	3	LOW	106.2866
150	51	2	MEDIUM	114.7136
143	58	0	HIGH	128.0429
144	30	3	MEDIUM	112.4824
145	32	2	HIGH	124.6376
150	35	0	VERY HIGH	164.926
128	83	3	MEDIUM	108.0024
133	86	2	MEDIUM	112.4782
141	89	0	HIGH	125.88
142	63	3	MEDIUM	108.6968
129	67	2	MEDIUM	110.1976
133	70	0	HIGH	125.735
141	37	3	MEDIUM	111.4594
142	39	2	HIGH	120.2391
129	42	0	HIGH	121.8082
113	76	3	LOW	99.33321
114	79	2	LOW	99.16275
123	82	0	MEDIUM	112.4728
126	46	3	LOW	103.7978
127	51	2	LOW	108.2963
114	58	0	MEDIUM	110.2368
123	30	3	LOW	98.96108
126	32	2	LOW	105.8817

MIX	ST	ROLL	PMT	PMT
127	35	0	MEDIUM	116.9455
98	83	3	LOW	82.07769
99	86	2	LOW	92.46046
102	89	0	LOW	98.9629
111	63	3	LOW	99.11745
112	67	2	LOW	99.30645
99	70	0	LOW	91.94335
102	37	3	LOW	99.05306
111	39	2	LOW	99.11745
112	42	0	LOW	105.6324
93	76	3	VERY LOW	71.24473
95	79	2	VERY LOW	71.13955
96	82	0	VERY LOW	73.90363
97	46	3	VERY LOW	77.09863
97.4	51	2	VERY LOW	77.79744
95	58	0	LOW	99.01753
96	30	3	VERY LOW	73.06663
97	32	2	VERY LOW	76.00736
97.4	35	0	LOW	99.35382

Table 7.13 Results for GBD

INPUT				DESIRED OUTPUT RANGE	OUTPUT
FP	PMT	PITCH	GRAPHITE	GBD	GBD
1501	132.6	4.73	6.7	LOW	2.992666
1510	133	4.74	5.6	MEDIUM	3.048886
1520	134	4.75	4.6	MEDIUM	3.118167
1530	135	4.76	3.5	HIGH	3.152436
1540	138	4.77	3	VERY HIGH	3.186247
1550	133.5	4.16	6.8	LOW	2.970636
1560	118	4.18	5.7	LOW	3.041819
1570	120	4.2	4.7	MEDIUM	3.066464
1580	122	4.61	3.7	HIGH	3.13335
1590	128	4.63	3.1	VERY HIGH	3.170482
1501	131	3.5	6.9	VERY LOW	2.963273
1510	132	3.8	5.8	LOW	3.031429
1520	122	4.09	5.2	MEDIUM	3.057368
1530	128	4.06	4.2	MEDIUM	3.12344
1540	131	4	3.2	HIGH	3.141707

FP	PMT	PITCH	GRAPHITE	GBD	GBD
1550	105.5	4.73	7	LOW	2.957423
1560	107	4.74	6.3	LOW	3.009607
1570	109	4.75	5.3	MEDIUM	3.064444
1580	112	4.76	4.4	HIGH	3.125822
1590	116	4.77	3.3	HIGH	3.130237
1501	118	4.16	6.7	VERY LOW	2.980307
1510	105.5	4.18	5.6	LOW	3.015385
1520	107	4.2	4.6	MEDIUM	3.059258
1530	109	4.61	3.5	MEDIUM	3.123885
1540	112	4.63	3	HIGH	3.149979
1550	116	3.5	6.8	VERY LOW	2.969346
1560	118	3.8	5.7	LOW	3.033553
1570	107	4.09	4.7	MEDIUM	3.057368
1580	109	4.06	3.7	MEDIUM	3.085443
1590	112	4	3.1	HIGH	3.123306
1501	93	4.73	6.9	VERY LOW	2.956708
1510	94	4.74	5.8	LOW	2.953998
1520	97	4.75	5.2	LOW	3.033318
1530	101	4.76	4.2	MEDIUM	3.106633
1540	103	4.77	3.2	MEDIUM	3.116753
1550	104	4.16	7	VERY LOW	2.956708
1560	88	4.18	6.3	LOW	2.954414
1570	92	4.2	5.3	MEDIUM	3.023861
1580	91	4.61	4.4	MEDIUM	3.064201
1590	92.4	4.63	3.3	HIGH	3.119332
1520	91.4	3.5	6.7	VERY LOW	2.955084
1530	90.5	3.8	5.6	LOW	2.979397
1540	104	4.09	4.6	LOW	3.05156
1550	88	4.06	3.5	MEDIUM	3.056876
1560	92	4	3	MEDIUM	3.101589
1480	132.6	4.73	6.8	LOW	2.992666
1225	133	4.74	5.7	LOW	3.003476
1250	134	4.75	4.7	MEDIUM	3.069755
1275	135	4.76	3.7	HIGH	3.13174
1300	138	4.77	3.1	VERY HIGH	3.15134
1325	133.5	4.16	6.9	VERY LOW	2.965709
1350	118	4.18	5.8	LOW	3.02424
1375	120	4.2	5.2	MEDIUM	3.055786
1400	122	4.61	4.2	HIGH	3.133247
1450	128	4.63	3.2	VERY HIGH	3.167437

FP	PMT	PITCH	GRAPHITE	GBD	GBD
1480	131	3.5	7	VERY LOW	2.953998
1225	132	3.8	6.3	LOW	2.990744
1250	122	4.09	5.3	LOW	3.04189
1275	128	4.06	4.4	MEDIUM	3.090102
1300	131	4	3.3	HIGH	3.106414
1325	105.5	4.73	6.7	VERY LOW	2.985729
1350	107	4.74	5.6	LOW	3.026371
1375	109	4.75	4.6	MEDIUM	3.102378
1400	112	4.76	3.5	HIGH	3.125322
1450	116	4.77	3	HIGH	3.150395
1480	118	4.16	6.8	VERY LOW	2.97379
1225	105.5	4.18	5.7	LOW	3.001776
1250	107	4.2	4.7	MEDIUM	3.031452
1275	109	4.61	3.7	MEDIUM	3.109506
1300	112	4.63	3.1	HIGH	3.135736
1325	116	3.5	6.9	VERY LOW	2.965709
1350	118	3.8	5.8	LOW	3.02424
1375	107	4.09	5.2	LOW	3.05356
1400	109	4.06	4.2	MEDIUM	3.08397
1450	112	4	3.2	MEDIUM	3.12105
1480	93	4.73	7	VERY LOW	2.956708
1225	94	4.74	6.3	VERY LOW	2.958171
1250	97	4.75	5.3	LOW	3.015207
1275	101	4.76	4.4	MEDIUM	3.069406
1300	103	4.77	3.3	MEDIUM	3.088405
1325	104	4.16	6.7	VERY LOW	2.983756
1350	88	4.18	5.6	VERY LOW	2.979397
1375	92	4.2	4.6	LOW	3.044582
1400	91	4.61	3.5	MEDIUM	3.094657
1450	92.4	4.63	3	MEDIUM	3.100014
1275	91.4	3.5	6.8	VERY LOW	2.957244
1300	90.5	3.8	5.7	VERY LOW	2.970449
1325	104	4.09	4.7	LOW	3.039333
1350	88	4.06	3.7	LOW	3.056401
1375	92	4	3.1	MEDIUM	3.079747
1005	132.6	4.73	6.9	VERY LOW	2.969589
1025	133	4.74	5.8	LOW	2.991839
1050	134	4.75	5.2	MEDIUM	3.059725
1075	135	4.76	4.2	MEDIUM	3.104344
1100	138	4.77	3.2	HIGH	3.136666

FP	PMT	PITCH	GRAPHITE	GBD	GBD
1125	133.5	4.16	7	VERY LOW	2.953731
1150	118	4.18	6.3	VERY LOW	2.97379
1200	120	4.2	5.3	LOW	3.037381
1225	122	4.61	4.4	MEDIUM	3.085698
1249	128	4.63	3.3	HIGH	3.134079
1005	131	3.5	6.7	VERY LOW	2.955084
1025	132	3.8	5.6	VERY LOW	2.981703
1050	122	4.09	4.6	LOW	3.048886
1075	128	4.06	3.5	LOW	3.071416
1100	131	4	3	MEDIUM	3.122014
1125	105.5	4.73	6.8	VERY LOW	2.978687
1150	107	4.74	5.7	LOW	3.000506
1200	109	4.75	4.7	MEDIUM	3.066082
1225	112	4.76	3.7	HIGH	3.125
1249	116	4.77	3.1	HIGH	3.127895
1005	118	4.16	6.9	VERY LOW	2.955334
1025	105.5	4.18	5.8	VERY LOW	2.957423
1050	107	4.2	5.2	LOW	3.009607
1075	109	4.61	4.2	MEDIUM	3.108503
1100	112	4.63	3.2	MEDIUM	3.123207
1125	116	3.5	7	VERY LOW	2.953391
1150	118	3.8	6.3	VERY LOW	2.97379
1200	107	4.09	5.3	LOW	3.02756
1225	109	4.06	4.4	MEDIUM	3.050326
1249	112	4	3.3	MEDIUM	3.125
1005	93	4.73	6.7	VERY LOW	2.956708
1025	94	4.74	5.6	VERY LOW	2.979397
1050	97	4.75	4.6	LOW	3.048886
1075	101	4.76	3.5	LOW	3.062264
1100	103	4.77	3	MEDIUM	3.116753
1125	104	4.16	6.8	VERY LOW	2.972683
1150	88	4.18	5.7	VERY LOW	2.966595
1200	92	4.2	4.7	LOW	3.005738
1225	91	4.61	3.7	MEDIUM	3.055786
1249	92.4	4.63	3.1	MEDIUM	3.076676
1005	91.4	3.5	6.9	VERY LOW	2.953425
1025	90.5	3.8	5.8	VERY LOW	2.951581
1050	104	4.09	5.2	VERY LOW	2.991889
1075	88	4.06	4.2	LOW	3.056069
1100	92	4	3.2	LOW	3.053844

Table 7.14 Results for COKED POROSITY

INPUT		OUTPUT
PITCH	GBD	CP
4.75	3.25	9.921306
4.6	3.23	9.695921
3.8	3.21	10.99956
4.8	3.19	8.655476
4.1	3.15	9.999726
4	3.13	10.06729
4.7	3.11	8.000736
4.4	3.09	9.000149
3.4	3.08	9.000049
4.72	3.07	8.000121
4.65	3.05	8.078939
4.05	2.98	7.63926
5	2.97	7.000507
4.4	2.95	7.000495
3.6	2.94	7.000495
4.72	3.21	9.260016
4.63	3.23	9.668509
4.2	3.25	10.9997
4.79	3.126	8.000749
4.16	3.15	9.905158
4.02	3.174	10.09738
4.8	3.076	7.985088
4.4	3.1	9.000099
3.8	3.128	10.09602
4.73	2.976	7.580277
4.19	3	8.000297
3.5	3.085	9.000033
5	2.974	7.000473
4.4	2.95	7.000495
3.9	2.8	7.000495

Table 7.15 Results for CCS

INPUT			DESIRED OUTPUT RANGE	OUTPUT
GBD	PITCH	TEMPERING	CCS	CCS
3.21	4.71	302	VERY_HIGH	466.432744
3.23	4.72	275	VERY_HIGH	492.947782
3.25	4.73	338	HIGH	505.704467
3.21	4.11	308	VERY_HIGH	472.313108
3.23	4.13	280	HIGH	413.754926
3.25	4.15	340	HIGH	508.453327
3.21	3.8	312	HIGH	407.54717
3.23	4.09	285	MEDIUM	421.408036
3.25	4.06	342	VERY_HIGH	510.077258
3.126	4.74	314	VERY_HIGH	459.667396
3.15	4.75	290	HIGH	435.302485
3.174	4.76	344	HIGH	430.78744
3.126	4.61	318	HIGH	459.667396
3.15	4.63	295	MEDIUM	444.56238
3.174	4.65	347	MEDIUM	412.456162
3.126	4.06	321	MEDIUM	376.285315
3.15	4	277	MEDIUM	376.923077
3.174	4.09	349	MEDIUM	400.9446
3.076	4.77	325	HIGH	400
3.1	4.78	282	HIGH	404.363636
3.128	4.79	344	MEDIUM	409.952273
3.076	4.15	328	MEDIUM	341.581549
3.1	4.16	286	MEDIUM	364.344262
3.128	4.18	339	LOW	374.883191
3.076	3.3	330	LOW	283.269231
3.1	3.4	289	MEDIUM	361.397671
3.128	3.5	345	LOW	353.274336
2.976	4.77	334	MEDIUM	294.219701
3	4.78	292	MEDIUM	326.877868
3.085	4.79	344	LOW	365.276983
2.976	4.63	327	MEDIUM	284.9803
3	4.65	294	MEDIUM	324.507937
3.085	4.69	340	LOW	369.005263
2.976	3.8	319	LOW	251.68468
3	4.09	297	LOW	317.873079

GBD	PITCH	TEMPERING	CCS	CCS
3.085	4.06	338	VERY_LOW	328.140996
2.974	4.74	320	VERY_LOW	283.318524
2.95	4.75	299	LOW	269.136259
2.8	4.76	341	VERY_LOW	239.834436
2.974	4.13	306	VERY_LOW	294.034087
2.95	4.15	296	LOW	266.993964
2.8	4.16	346	VERY_LOW	238.417754
2.974	3.3	311	VERY_LOW	253.458215
2.95	3.4	283	VERY_LOW	238.417754
2.8	3.5	344	VERY_LOW	238.983107

Table 8.4 Results for 86 data set obtained using Matlab (for trimf & trapf) Vs Prolog based System

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLLS	FP	FP	GBD	GBD MATLAB TRIMF	DIFFERENCE	GBD MATLAB TRAPF	DIFFERENCE	GBD PROLAOG	DIFFERENCE
319.5	574.5	5	129	8	134	132	55	0	195	1418.18	3.1	3.08	-0.02	3.19	0.09	3.11	0.01
248.6	351.5	3	122	8	124	105	90	1	195	1418.18	3.06	3.02	-0.04	3.05	-0	3.07	0.01
359	485	6	107	14	115	112	30	0	190	1381.82	3.01	3.03	0.02	2.5	-0.5	3.03	0.02
276	420.1	7	125	11	131	122	35	1	204	1483.64	3.13	3.13	-0	3.15	0.02	3.14	0.01
272	445	7	126	13	126	120	175	0	210	1527.27	3.11	3.11	-0	3.11	0	3.12	0.01
386	599	4	127	14	135	122	48	1	212	1541.82	3.11	3.11	-0	3.1	-0	3.11	0
361	570	4	130	9	130	120	108	0	200	1454.55	3.08	3.09	0.01	3.08	0	3.07	0.01
388.8	435.8	5	127	31.5	128	115	20	1	193	1403.64	3.11	3.11	-0	3.11	-0	3.11	0
370	428	7	130	33	148	122	65	1	193	1403.64	3.04	3.05	0.01	3.07	0.03	3.03	0.01
423	500	5.8	127	7	128	128	31	0	211	1534.55	3.09	3.09	0	3.22	0.13	3.11	0.02
440	579	5.3	104	8	135	115	26	1	194	1410.91	3.12	3.12	0	3.1	-0	3.11	0.01
438	539	4	106	10	140	128	20	1	212	1541.82	3.09	3.09	0	3.09	-0	3.11	0.02
416	494	4	128	11	135	125	30	1	210	1527.27	3.09	3.1	0.01	3.09	0	3.11	0.02
416	514	4	131	11	140	127	45	1	211	1534.55	3.09	3.08	-0.01	3.09	0	3.11	0.02
380	440	4	123	11	135	123	45	1	210	1527.27	3.08	3.08	0	3.1	0.02	3.09	0.01
309	340	5	105	11	130	122	33	0	214	1556.36	3.14	3.14	-0	3.14	-0	3.13	0.01
286	314	6	108	8	130	117	42	1	211	1534.55	3.06	3.06	-0	3.08	0.02	3.07	0.01
265	290	7	105	8	120	120	43	0	212	1541.82	3.11	3.11	0	3.11	0	3.09	0.02
288	301	6	105	9	120	120	38	0	209	1520	3.08	3.08	-0	3.07	-0	3.07	0.01
518	489	5	118	8	140	128	115	0	195	1418.18	3.02	3.02	0	3.06	0.04	3.03	0.01
454	389	5	132	5	150	130	135	1	194	1410.91	3.06	3.05	-0.01	3.04	-0	3.07	0.01
403	516	4	106	7	133	129	50	0	194	1410.91	3.06	3.1	0.04	3.06	0	3.09	0.03
395	499	4	111	7	132	127	75	0	195	1418.18	3.09	3.09	-0	3.07	-0	3.09	0
388	395	5	120	11	125	125	35	0	193	1403.64	3.08	3.1	0.02	3.12	0.04	3.07	0.01
376	420	6	124	9	130	123	25	1	195	1418.18	3.13	3.14	0.01	3.11	-0	3.14	0.01
408	405	6	128	5	127	126	35	0	195	1418.18	3.11	3.11	-0	3.11	0	3.11	0
392	407	5	128	5	122	122	25	0	197	1432.73	3.12	3.12	-0	3.11	-0	3.11	0.01
384	415	5	122	5	125	125	40	0	196	1425.45	3.11	3.1	-0.01	3.12	0.01	3.12	0.01
371.2	434.3	7	120	8	140	124	175	0	210	1527.27	3.1	3.1	0	3.1	0	3.11	0.01
350	717	5	113	14	112	112	20	0	195	1418.18	3.09	3.07	-0.02	3.11	0.02	3.11	0.02
393.3	437	4	125	20	144	125	170	0	195	1418.18	3.03	3.03	0	3.06	0.03	3.03	0
555.4	436.3	5	134	10	178	127	180	1	194	1410.91	3.05	3.04	-0.01	3	-0	3.04	0.01
255	398	10	115	8	128	128	21	0	195	1418.18	3.14	3.12	-0.02	3.11	-0	3.15	0.01
287	560	8.5	115	6.25	166	129	25	1	195	1418.18	3.13	3.13	-0	3.12	-0	3.14	0.01
307	630	6	119	7.5	124	124	32	0	193	1403.64	3.11	3.12	0.01	3.13	0.02	3.11	0
293	505	5	114	9	116	116	80	0	193	1403.64	3.05	3.05	0	3.05	-0	3.06	0.01
345.7	685.9	4.5	126	8	130	125	10	1	195	1418.18	3.1	3.1	-0	3.11	0.01	3.11	0.01
333.8	658	4	126	11	128	125	20	0	196	1425.45	3.15	3.12	-0.03	3.11	-0	3.13	0.02
276	453.5	4.5	123	8	118	118	40	0	194	1410.91	3.14	3.15	0.01	3.12	-0	3.14	0

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLLS	FP	FP	GBD	GBD MATLAB TRIMF	DIFFERENCE	GBD MATLAB TRAPF	DIFFERENCE	GBD PROLAOG	DIFFERENCE
248.7	417.1	5	121	8	110	109	186	0	197	1432.73	3.1	3.09	-0.01	3.1	0	3.11	0.01
304	517	6	120	6	145	125	25	1	194	1410.91	3.11	3.06	-0.05	3.21	0.1	3.11	0
286	493	8	126	8	125	124	185	0	196	1425.45	3.09	3.1	0.01	3.11	0.02	3.11	0.02
291	513	5	124	5	135	125	145	0	195	1418.18	3.13	3.13	-0	3.06	-0.1	3.14	0.01
284	506.6	7	91	8.5	123	114	180	0	193	1403.64	3.12	3.12	0	3.12	0	3.12	0
314	717	5	130	9	135	120	47	1	194	1410.91	3.13	3.12	-0.01	3.09	-0	3.14	0.01
340	441	8	119	18	135	130	37	0	185	1345.45	3.17	3.17	-0	3.17	-0	3.18	0.01
321.2	425.1	8	124	8	130	125	31	0	218	1585.45	3.15	3.16	0.01	3.17	0.02	3.14	0.01
330	408	6	122	12	116	114	20	0	188	1367.27	3.12	3.13	0.01	3.08	-0	3.12	0
380	520	6	118	16	140	128	13	1	192	1396.36	3.11	3.11	0	3.83	0.72	3.11	0
417	602	6	118	9	143	130	25	1	192	1396.36	3.1	3.09	-0.01	3.12	0.02	3.11	0.01
401	580	5	124	8	134	120	43	1	195	1418.18	3.15	3.15	0	3.14	-0	3.14	0.01
389	567	5	125	11	135	122	17	1	196	1425.45	3.14	3.14	0	3.11	-0	3.13	0.01
320	410	6	117	8	120	118	27	0	192	1396.36	3.16	3.19	0.03	3.12	-0	3.13	0.03
292	370	7	119	11	125	121	50	0	199	1447.27	3.18	3.14	-0.04	3.13	-0.1	3.15	0.03
285	366	6	119	10	120	114	78	0	202	1469.09	3.15	3.15	-0	2.86	-0.3	3.14	0.01
396	536	8	124	12	157	130	25	1	203	1476.36	3.17	3.18	0.01	3.14	-0	3.18	0.01
354	552	7	112	14	115	110	20	0	201	1461.82	3.15	3.16	0.01	2.52	-0.6	3.14	0.01
272	482	9	116	10	115	114	6	0	198	1440	3.11	3.11	0	3.11	0	3.11	0
278	480	7	117	9	113	111	13	0	201	1461.82	3.09	3.09	0	2.29	-0.8	3.12	0.03
271	472	7	115	12	115	112	26	0	203	1476.36	3.13	3.13	0	3.11	-0	3.13	0
296	524	9	114	7	127	125	29	0	194	1410.91	3.08	3.1	0.02	3.12	0.04	3.08	0
285	509	9	120	10	125	124	45	0	201	1461.82	3.09	3.1	0.01	3.12	0.03	3.1	0.01
303	530	8	115	9	122	118	78	0	199	1447.27	3.13	3.15	0.02	3.15	0.02	3.12	0.01
418	556	7	113	8	135	125	30	1	204	1483.64	3.1	3.1	-0	3.11	0.01	3.12	0.02
400	544	7	117	8	137	126	30	1	185	1345.45	3.16	3.16	0	3.3	0.14	3.13	0.03
396	548	6	119	8	127	127	15	0	204	1483.64	3.13	3.13	0	3.12	-0	3.14	0.01
380	490	6	121	9	119	119	10	0	183	1330.91	3.17	3.14	-0.03	3.11	-0.1	3.18	0.01
338	418	8	106	10	148	142	15	1	204	1483.64	3.12	3.12	0	3.11	-0	3.11	0.01
329	428	7	124	11	125	115	10	1	204	1483.64	3.17	3.2	0.03	3.13	-0	3.18	0.01
344	428	7	117	11	140	118	57	1	185	1345.45	3.12	3.02	-0.1	3.12	0	3.12	0
344	421	8	121	17	124	120	17	1	204	1483.64	3.05	3.04	-0.01	3.05	0	3.03	0.02
323	401	9	126	18	150	137	5	1	191	1389.09	3.09	3.09	0	3.09	-0	3.11	0.02
307	392	8	130	19	148	142	13	1	199	1447.27	3.11	3.02	-0.09	3.11	0	3.11	0
308	410	8	130	19	143	135	75	0	195	1418.18	3.08	3.08	0	3.08	-0	3.07	0.01
323	392	7	118	9	136	125	67	1	190	1381.82	3.14	3.12	-0.02	3.11	-0	3.14	0
334	467	6	120	8	137	127	85	1	190	1381.82	3.09	3.01	-0.08	3.6	0.51	3.12	0.03
376	505	5	117	8	135	124	72	1	190	1381.82	3.03	2.92	-0.11	3.4	0.37	3.08	0.05
356	487	6	123	8	138	125	79	1	190	1381.82	3.13	3.14	0.01	3.18	0.05	3.13	0
368	494	5	125	10	136	124	72	0	190	1381.82	3.14	3.14	-0	3.14	-0	3.14	0
377	507	6	127	7	140	128	91	0	190	1381.82	3.13	3.13	-0	3.14	0.01	3.14	0.01
390	501	5	125	9	142	129	62	0	190	1381.82	3.12	3.18	0.06	3.1	-0	3.12	0
383	583	5	128	8	138	124	25	0	190	1381.82	3.12	3.11	-0.01	3.11	-0	3.11	0.01

CTI	GHDT	RT	MIXER TEMP	MIXING TIME	MIX TEMP	PMT	ST	NO OF ROLLS	FP	FP	GBD	GBD MATLAB TRIMF	DIFFERENCE	GBD MATLAB TRAPF	DIFFERENCE	GBD PROLAOG	DIFFERENCE
391	506	5	122	11	130	118	15	1	195	1418.18	3.1	3.09	-0.01	3.11	0.01	3.1	0
394	522	4	124	5	137	121	30	1	198	1440	3.11	3.11	-0	3.1	-0	3.11	0
389	516	5	121	6	127	125	5	0	203	1476.36	3.06	3	-0.06	3.12	0.06	3.08	0.02
375	501	4	126	4	128	126	5	0	196	1425.45	3.13	3.15	0.02	3.11	-0	3.14	0.01

Table 8.5 Results for 117 Data Set Obtained Using Matlab For Mix Temperature And GBD Using Triangular & Trapezoidal Membership Functions

CTI	GHDT	RT	MIXER TEMP	mixing time	mix temp	mix temp using matlab trimf	difference	mix temp using matlab trapf	difference	pmt	st	no of roll	fp	gbd	gbd using matlab trimf	difference	gbd using matlab trapf	difference
319.5	574.5	5	129	8	134	132.5	2	134	-0	132	55	0	1418	3.1	3.11	-0.01	3.08	-0
248.6	351.5	3	122	8	124.4	124.1	0	124	-0	104.7	90	1	1418	3.06	3.07	-0.01	3.05	-0
359	485	6	107	14	115	115	-0	135	20	112	30	0	1382	3.01	3.03	-0.02	10.9	7.91
276	420.1	7	125	11	131	127.3	4	127	-4	122	35	1	1484	3.13	3.16	-0.03	3.14	0.01
272	445	7	126	13	126	126.3	-0	125	-1	120	175	0	1527	3.11	3.11	0	3.11	-0
386	599	4	127	14	135	135.1	-0	135	-0	122	48	1	1542	3.11	3.09	0.02	3.1	-0
361	570	4	130	9	130	129.3	1	130	0.4	120	108	0	1455	3.08	3.11	-0.03	3.11	0.03
388.8	435.8	5	126.5	31.5	128	128	-0	128	-0	115	20	1	1404	3.11	3.11	0	3.11	-0
370	428	7	130	33	148	148	-0	148	-0	122	65	1	1404	3.04	3.04	0	3.09	0.05
423	500	5.75	127	7	128	128.1	-0	136	7.6	128	31	0	1535	3.09	3.09	0	3.2	0.11
440	579	5.25	104	8	135	135	-0	135	-0	115	26	1	1411	3.12	3.11	0.01	3.12	-0
438	539	4	106	10	140	140	0	140	-0	128	20	1	1542	3.09	3.1	-0.01	3.09	-0
416	494	4	128	11	135	135.5	-0	135	0.4	125	30	1	1527	3.09	3.06	0.03	3.09	-0
416	514	4	131	11	140	139.8	0	137	-3	127	45	1	1535	3.09	3.09	-0	3.09	0
380	440	4	123	11	135	134.8	0	131	-4	123	45	1	1527	3.08	3.11	-0.03	3.39	0.31
309	340	5	105	11	130	130	0	130	-0	122	33	0	1556	3.14	3.14	0	3.14	0
286	314	6	108	8	130	130	0	129	-1	117	42	1	1535	3.06	3.06	0	3.19	0.13
265	290	7	105	8	120	120	-0	120	-0	120	43	0	1542	3.11	3.11	0	3.11	0
288	301	6	105	9	120	120	-0	120	0	120	38	0	1520	3.08	3.08	-0	3.07	-0
518	489	5	118	8	140	140	0	140	-0	128	115	0	1418	3.02	3.02	-0	3.1	0.08
454	389	5	132	5	150	150.1	-0	150	-0	130	135	1	1411	3.06	3.06	-0	3.09	0.03
403	516	4	106	7	133	132.8	0	133	0	129	50	0	1411	3.06	3.09	-0.03	3.09	0.03
395	499	4	111	7	132	132.5	-0	133	0.7	127	75	0	1418	3.09	3.06	0.03	3.09	0
388	395	5	120	11	125	125.2	-0	125	0	125	35	0	1404	3.08	3.1	-0.02	3.11	0.03
376	420	6	124	9	130	129.3	1	132	2	123	25	1	1418	3.13	3.13	-0	3.11	-0
408	405	6	128	5	127	126.8	0	126	-1	126	35	0	1418	3.11	3.11	-0	3.11	0
392	407	5	128	5	122	122	-0	122	0	122	25	0	1433	3.12	3.13	-0.01	3.12	-0
384	415	5	122	5	125	124.9	0	125	-0	125	40	0	1425	3.11	3.11	-0	3.11	0
371.2	434.3	7	119.7	8	140	138.1	2	132	-8	124	175	0	1527	3.1	3.08	0.02	3.1	0
350	717	5	113	14	112	112	0	112	0	112	20	0	1418	3.09	3.09	-0	3.09	-0
393.3	437	4	125.3	20	144	144	0	144	-0	125	170	0	1418	3.03	3.04	-0.01	3.04	0.01
555.4	436.3	5	134	10	178	177.9	0	178	0	127	180	1	1411	3.05	3.06	-0.01	3.05	-0
255	398	10	115	8	128	127.8	0	128	-0	128	21	0	1418	3.14	3.13	0.01	3.13	-0

CTI	GHDT	RT	MIXER TEMP	mixing time	mix temp	mix temp using matlab trimf	difference	mix temp using matlab trapf	difference	pmt	st	no of roll	fp	gbd	gbd using matlab trimf	difference	gbd using matlab trapf	difference
287	560	8.5	115	6.25	166	166.1	-0	165	-1	129	25	1	1418	3.13	3.13	0	3.13	-0
307	630	6	119	7.5	124	123.7	0	124	-0	124	32	0	1404	3.11	3.12	-0.01	3.11	0
293	505	5	114	9	116	115.8	0	116	0	116	80	0	1404	3.05	3.06	-0.01	3.05	0
345.7	685.9	4.5	125.9	8	130	130.7	-1	130	-0	125	10	1	1418	3.1	3.1	-0	3.11	0.01
333.8	658	4	126	11	128	127.9	0	128	0	125	20	0	1425	3.15	3.11	0.04	3.11	-0
276	453.5	4.5	123	8	118	119.1	-1	118	-0	118	40	0	1411	3.14	3.16	-0.02	3.12	-0
248.7	417.1	5	120.5	8	110	110.8	-1	110	0.2	109	186	0	1433	3.1	3.1	0	3.1	0
304	517	6	120	6	145	147.4	-2	132	-13	125	25	1	1411	3.11	3.06	0.05	3.11	0
286	493	8	125.9	8	125	124.9	0	125	-0	124	185	0	1425	3.09	3.09	0	3.09	0
291	513	5	124.2	5	135	135.3	-0	135	0.5	125	145	0	1418	3.13	3.13	-0	3.09	-0
284	506.6	7	91	8.5	123	123	0	123	0	114	180	0	1404	3.12	3.14	-0.02	3.12	-0
314	717	5	130	9	135	135.2	-0	135	0	120	47	1	1411	3.13	3.13	0	3.11	-0
340	441	8	118.7	18	135	133	2	135	0.4	130	37	0	1345	3.17	3.12	0.05	3.16	-0
321.2	425.1	8	123.9	8	130	131.7	-2	132	2	125	31	0	1585	3.15	3.21	-0.06	3.17	0.02
330	408	6	122	12	116	122.2	-6	126	10	114	20	0	1367	3.12	3.08	0.04	5.38	2.26
380	520	6	118	16	140	141.4	-1	135	-5	128	13	1	1396	3.11	3.11	0	3.09	-0
417	602	6	118	9	143	143.1	-0	143	-0	130	25	1	1396	3.1	3.1	-0	3.11	0.01
401	580	5	124	8	134	131	3	134	0.2	120	43	1	1418	3.15	3.18	-0.03	3.14	-0
389	567	5	125.2	11	135	136.9	-2	135	-0	122	17	1	1425	3.14	3.14	0	3.12	-0
320	410	6	117	8	120	121.5	-2	131	11	118	27	0	1396	3.16	3.16	-0	3.39	0.23
292	370	7	119	11	125	124.2	1	121	-4	121	50	0	1447	3.18	3.15	0.03	3.12	-0.1
285	366	6	119	10	120	118.3	2	124	4.1	114	78	0	1469	3.15	3.17	-0.02	3.56	0.41
396	536	8	124	12	157	156.7	0	154	-3	130	25	1	1476	3.17	3.17	0	3.11	-0.1
354	552	7	112.1	14	115	115	-0	134	19	110	20	0	1462	3.15	3.17	-0.02	10.4	7.25
272	482	9	115.6	10	115	115.6	-1	120	5	114	6	0	1440	3.11	3.12	-0.01	3.64	0.53
278	480	7	116.8	9	113	114.2	-1	128	15	111	13	0	1462	3.09	3.12	-0.03	7.64	4.55
271	472	7	114.6	12	115	113.3	2	110	-5	112	26	0	1476	3.13	3.1	0.03	3.13	0
296	524	9	114.2	7	127	126.3	1	128	1.1	125	29	0	1411	3.08	3.11	-0.03	3.11	0.03
285	509	9	119.5	10	125	124.5	0	120	-5	124	45	0	1462	3.09	3.1	-0.01	3.12	0.03
303	530	8	114.9	9	122	122.8	-1	120	-2	118	78	0	1447	3.13	3.12	0.01	3.12	-0
418	556	7	113	8	135	135	-0	133	-2	125	30	1	1484	3.1	3.09	0.01	3.1	-0
400	544	7	117	8	137	137.2	-0	133	-4	126	30	1	1345	3.16	3.17	-0.01	3.13	-0
396	548	6	119	8	127	127.4	-0	133	5.8	127	15	0	1484	3.13	3.14	-0.01	3.12	-0
380	490	6	121	9	119	126.5	-7	133	14	119	10	0	1331	3.17	2.83	0.34	2.88	-0.3
338	418	8	106	10	148	148	0	148	-0	142	15	1	1484	3.12	3.12	0	3.1	-0
329	428	7	124	11	125	129.5	-4	133	8	115	10	1	1484	3.17	3.15	0.02	2.95	-0.2
344	428	7	117	11	140	140.3	-0	130	-10	118	57	1	1345	3.12	3.14	-0.02	3.17	0.05
344	421	8	121	17	124	126.9	-3	126	1.9	120	17	1	1484	3.05	3.06	-0.01	3.06	0.01
323	401	9	126	18	150	150.8	-1	150	-0	137	5	1	1389	3.09	3.12	-0.03	3.1	0.01
307	392	8	130	19	148	146.3	2	148	-0	142	13	1	1447	3.11	3.04	0.07	3.09	-0
308	410	8	130	19	143	144.6	-2	143	0.4	135	75	0	1418	3.08	3.07	0.01	3.08	-0
323	392	7	118	9	136	138.2	-2	129	-7	125	67	1	1382	3.14	3.15	-0.01	3.88	0.74
334	467	6	119.5	8	137	127.9	9	133	-4	127	85	1	1382	3.09	2.96	0.13	3.43	0.34
376	505	5	117	8	135	129.5	6	133	-2	124	72	1	1382	3.03	2.97	0.06	3.4	0.37
356	487	6	123	8	138	137.3	1	134	-4	125	79	1	1382	3.13	3.12	0.01	3.29	0.16

CTI	GHDT	RT	MIXER TEMP	mixing time	mix temp	mix temp using matlab trimf	diff ere nce	mix temp using matlab trapf	diffe renc e	pmt	st	no of roll	fp	gbd	gbd using matlab trimf	differen ce	gbd using matlab trapf	differe nce
368	494	5	125	10	136	134.3	2	135	-1	124	72	0	1382	3.14	3.11	0.03	3.1	-0
377	507	6	127	7	140	139	1	136	-4	128	91	0	1382	3.13	3.12	0.01	3.1	-0
390	501	5	125	9	142	137.2	5	135	-7	129	62	0	1382	3.12	3.11	0.01	3.09	-0
383	583	5	128	8	138	138.8	-1	138	-0	124	25	0	1382	3.12	3.13	-0.01	3.1	-0
391	506	5	122	11	130	133.5	-4	133	2.9	118	15	1	1418	3.1	3.09	0.01	3.11	0.01
394	522	4	124	5	137	138.3	-1	134	-3	121	30	1	1440	3.11	3.11	0	3.11	0
389	516	5	121	6	127	131.8	-5	133	5.8	125	5	0	1476	3.06	3.06	0	3.11	0.05
375	501	4	126	4	128	128.1	-0	134	6.4	126	5	0	1425	3.13	3.12	0.01	3.1	-0
414	519	6	126	16	153	152.9	0	138	-15	132	85	1	1484	3.09	3.07	0.02	3.09	0
385	532	5	128	15	135	134.2	1	138	2.9	128	90	1	1353	3.07	3.06	0.01	3.07	-0
380	480	6	123	16	140	140.4	-0	136	-4	130	28	1	1345	3.11	3.1	0.01	3.09	-0
361	494	6	126	16	135	133	2	138	2.8	128	108	0	1345	3.05	2.98	0.07	3.05	0
403	511	6	120	14	145	142.2	3	135	-10	138	45	0	1484	3.06	3.08	-0.02	3.05	-0
364	502	6	124	14	125	131.5	-6	137	12	121	20	0	1345	3.05	3	0.05	2.93	-0.1
353	454	6	120	14	125	131.7	-7	133	8	120	20	0	1484	3.07	3.11	-0.04	3.12	0.05
365	466	6	116	14	140	131.1	9	135	-5	125	60	0	1396	3.05	3.06	-0.01	3.1	0.05
360	411	7	123	14	144	137.1	7	140	-4	144	10	1	1476	3.09	3.15	-0.06	3.1	0.01
365	381	9	120	12	137	136.4	1	137	-0	130	50	1	1360	3.06	3.06	0	3.06	-0
368	369	7	118	14	142	143.2	-1	142	-0	130	85	0	1484	3.06	3.07	-0.01	3.06	-0
380	490	7	105	16	135	135	0	134	-1	135	20	1	1484	3.08	3.09	-0.01	2.46	-0.6
364	492	6	135	15	153	148.2	5	140	-13	140	40	0	1491	3.07	3.16	-0.09	3.07	-0
359	484	7	133	16	143	144.6	-2	140	-3	135	210	0	1462	3.12	2.99	0.13	3.12	-0
404	505	6	127	16	136	137.7	-2	138	2.3	135	12	0	1258	3.14	3.16	-0.02	3.14	0
374	521	6	129	17	135	134.5	1	139	4.5	134	19	0	1353	3.1	3.1	-0	3.1	0
354	500	6	129	17	132	134	-2	139	6.7	131	12	0	1360	3.13	3.11	0.02	3.07	-0.1
364	539	6	129	16	134	132.9	1	139	5.5	132	28	0	1338	3.12	3.13	-0.01	3.04	-0.1
359	539	6	131	14	138	138.5	-1	140	2	135	31	0	1353	3.12	3.13	-0.01	3.12	0
373	587	6	129	14	132	132.5	-0	132	-0	128	128	1	1345	3.12	3.12	-0	3.12	0
460	380	7	125	16	145	145	-0	145	-0	135	30	1	1491	3.12	3.1	0.02	3.1	-0
357	466	8	131	16	146	147.4	-1	147	1.2	136	38	1	1484	3.06	3.08	-0.02	3.1	0.04
349	475	8	137	16	146	144.1	2	146	-0	138	83	1	1476	3.11	3.1	0.01	3.1	-0
346	472	8	135	20	145	145.1	-0	145	-0	140	90	2	1484	3.05	3.03	0.02	3.01	-0
382	528	6	118	18	160	159.7	0	135	-25	140	45	1	1476	3.15	3.15	0	2.98	-0.2
376	498	4	125	17	129	130	-1	131	2	125	5	1	1345	3.08	3.1	-0.02	3.11	0.03
395	490	7	123	28	135	135	0	135	-0	130	5	1	1353	3.08	3.08	0	3.09	0.01
377	510	7	129.6	26	140	140	-0	141	0.7	131	20	1	1345	3.04	3.05	-0.01	3.07	0.03
363	479	7	135	27	142	142	-0	141	-1	132	30	1	1345	3.08	3.08	0	3.06	-0
355	477	7	136	27	140	140	0	140	-0	130	40	1	1353	3.06	3.08	-0.02	3.07	0.01
329.8	614	5	120	7	120	119.5	1	120	0	115	76	1	1411	3.1	3.15	-0.05	3.11	0.01

MEMBERSHIP FUNCTIONS FOR MIX TEMPERATURE FIS

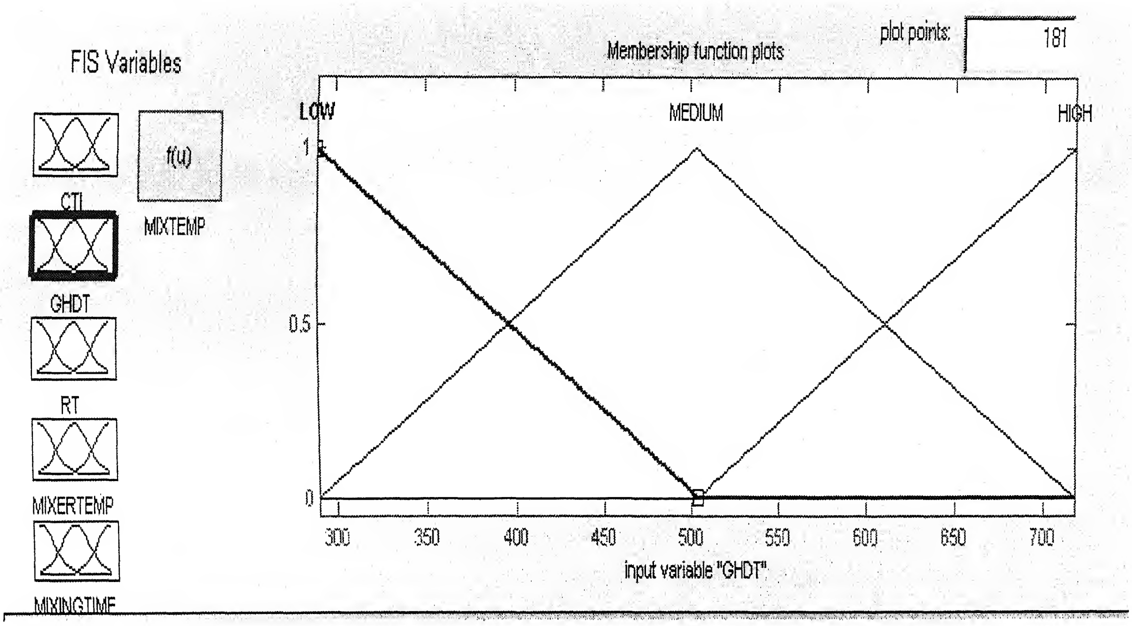


FIG 8.14 Membership function for GHDT

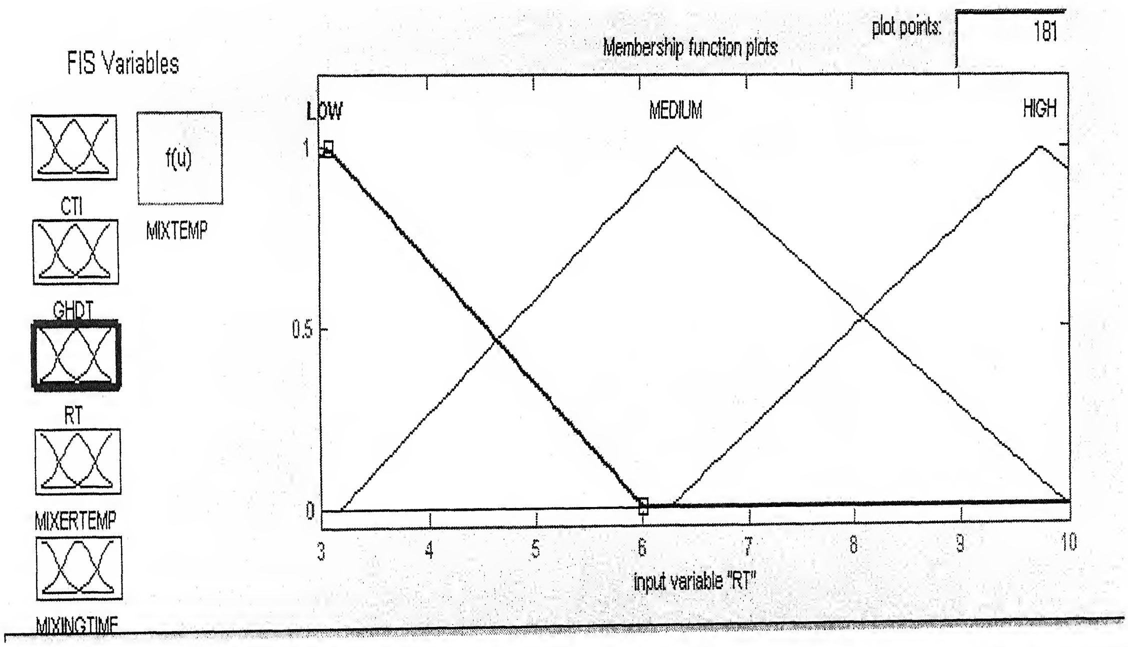


FIG 8.15 Membership function for RT

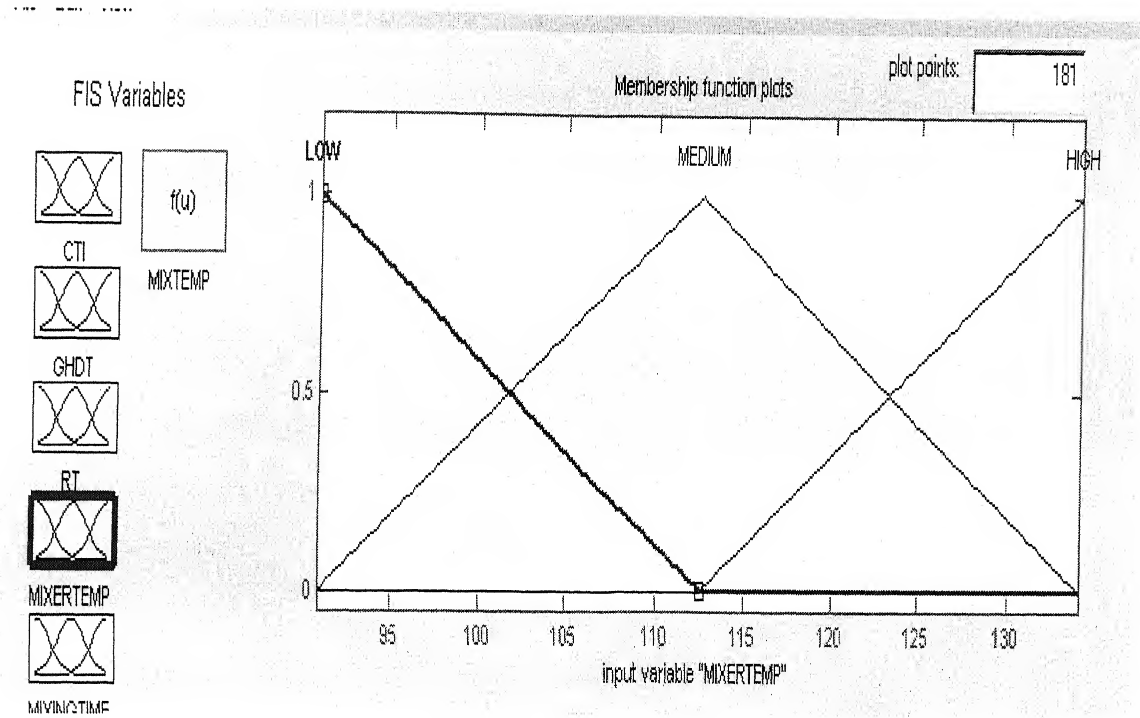


FIG 8.16 Membership function for Mixer Temperature

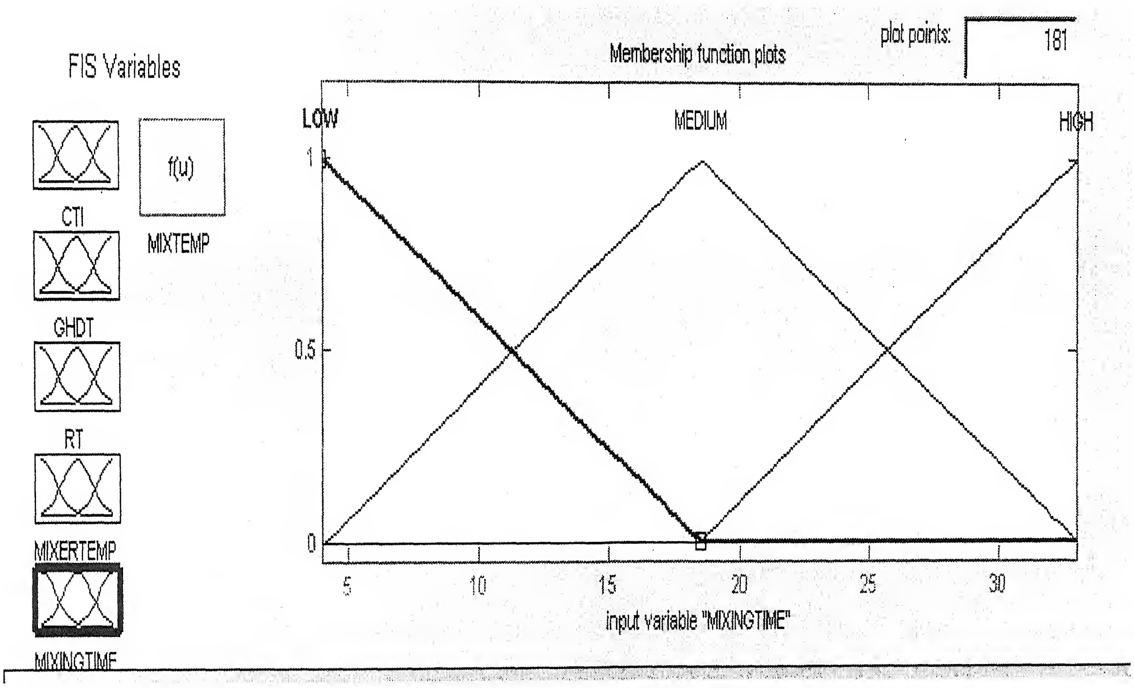


FIG 8.17 Membership function for Mixing Time

SURFACE PLOTS FROM MIX TEMPERATURE FIS

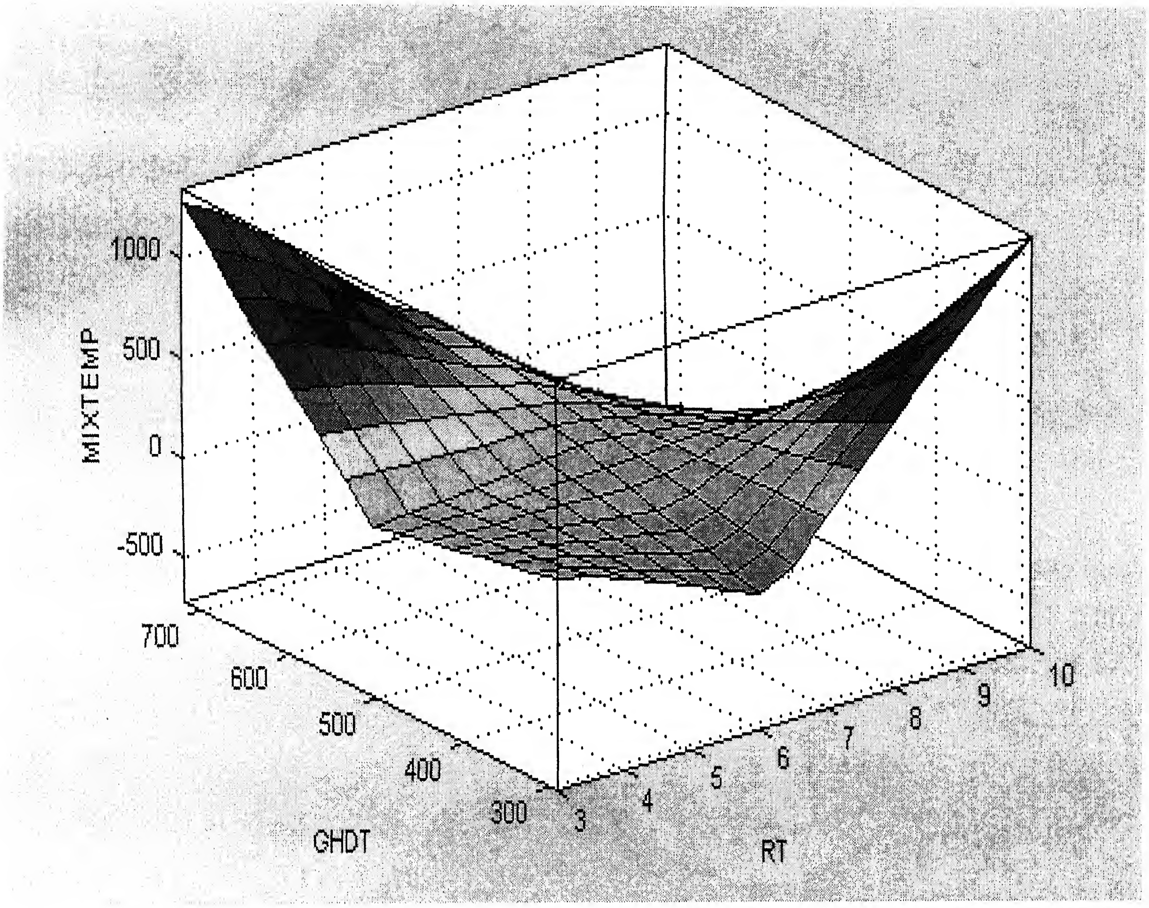


Fig 8.18 Surface Plot of RT & GHDT Vs Mix Temperature

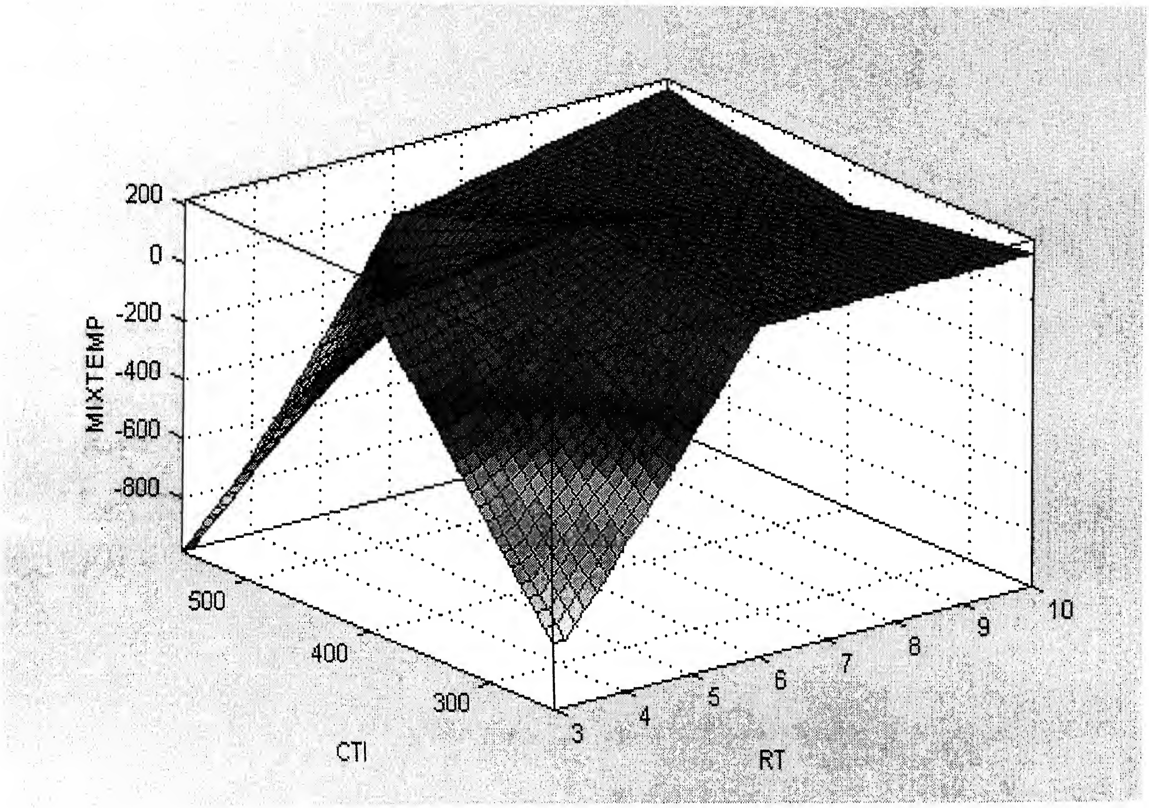


Fig 8.19 Surface Plot of RT & CTI Vs Mix Temperature

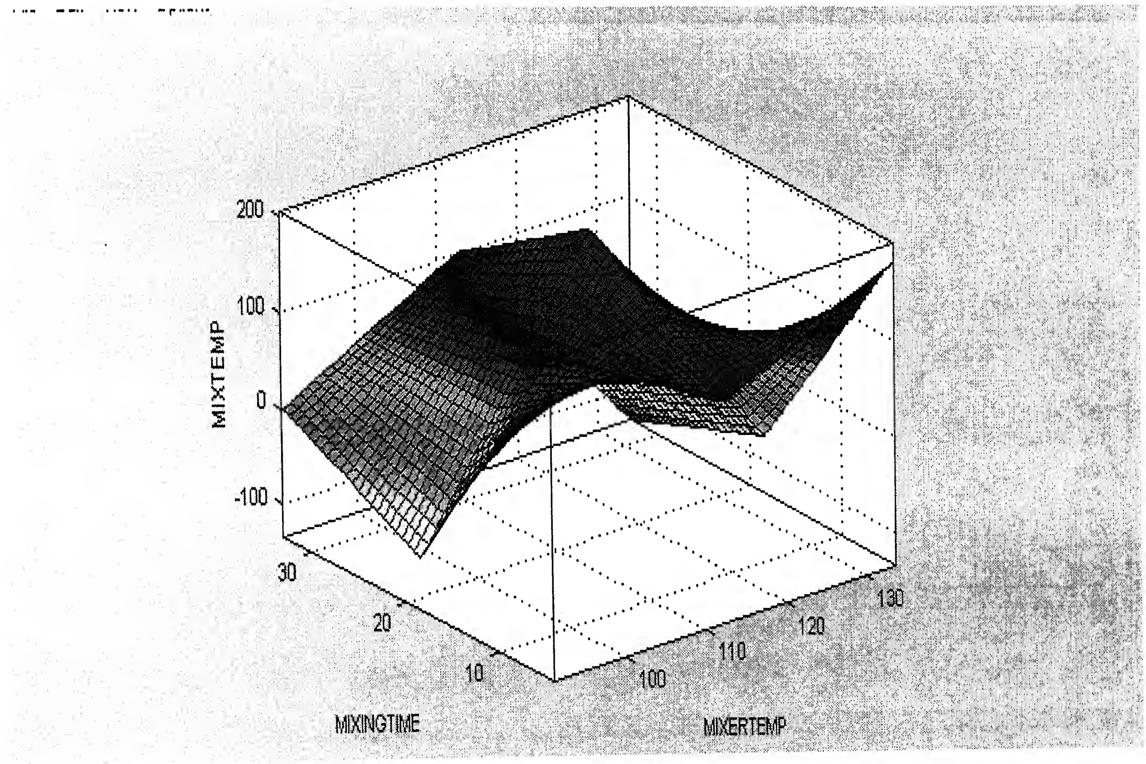


Fig 8.20 Surface Plot of Mixer Temperature & Mixing Time Vs Mix Temperature

FIS SYSTEM GENERATED FOR GREEN BULK DENSITY

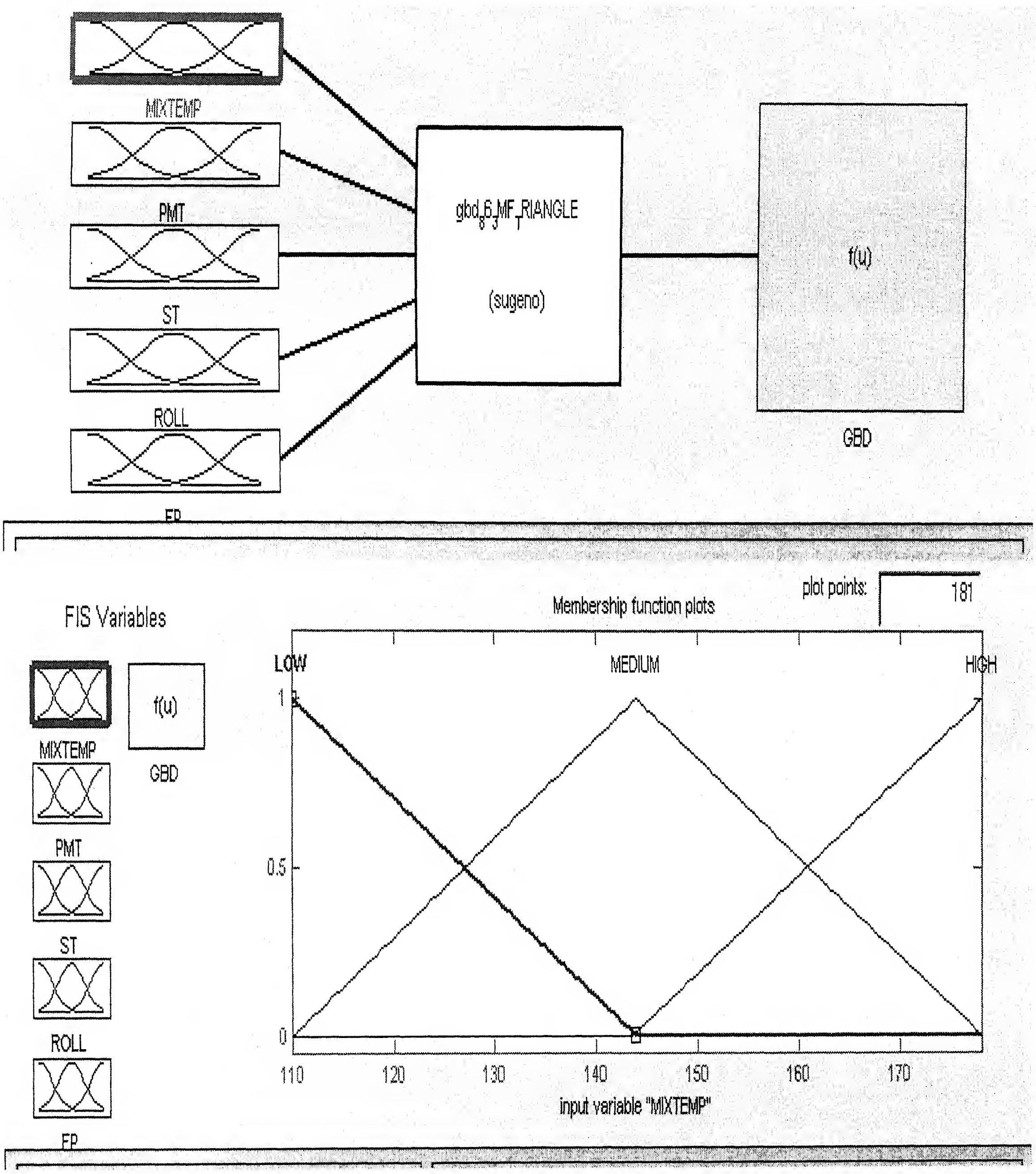
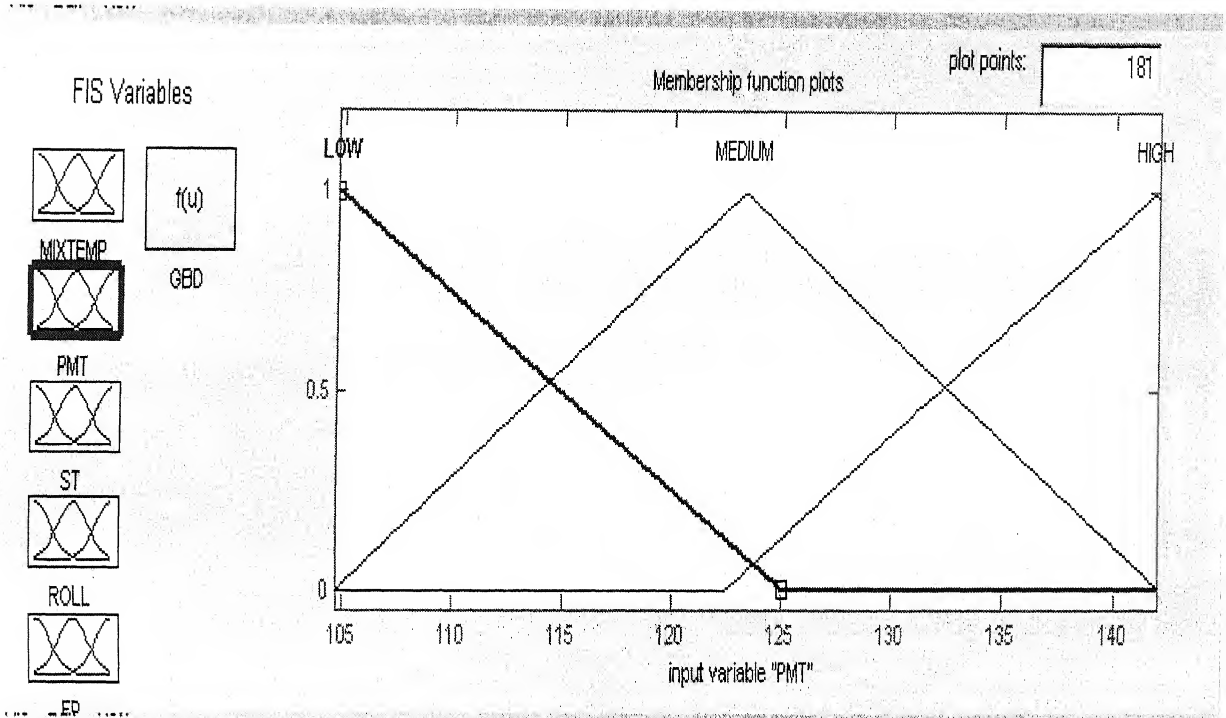


Fig 8.21 & Fig 8.22 Fis Structure Membership Function for Mix Temperature



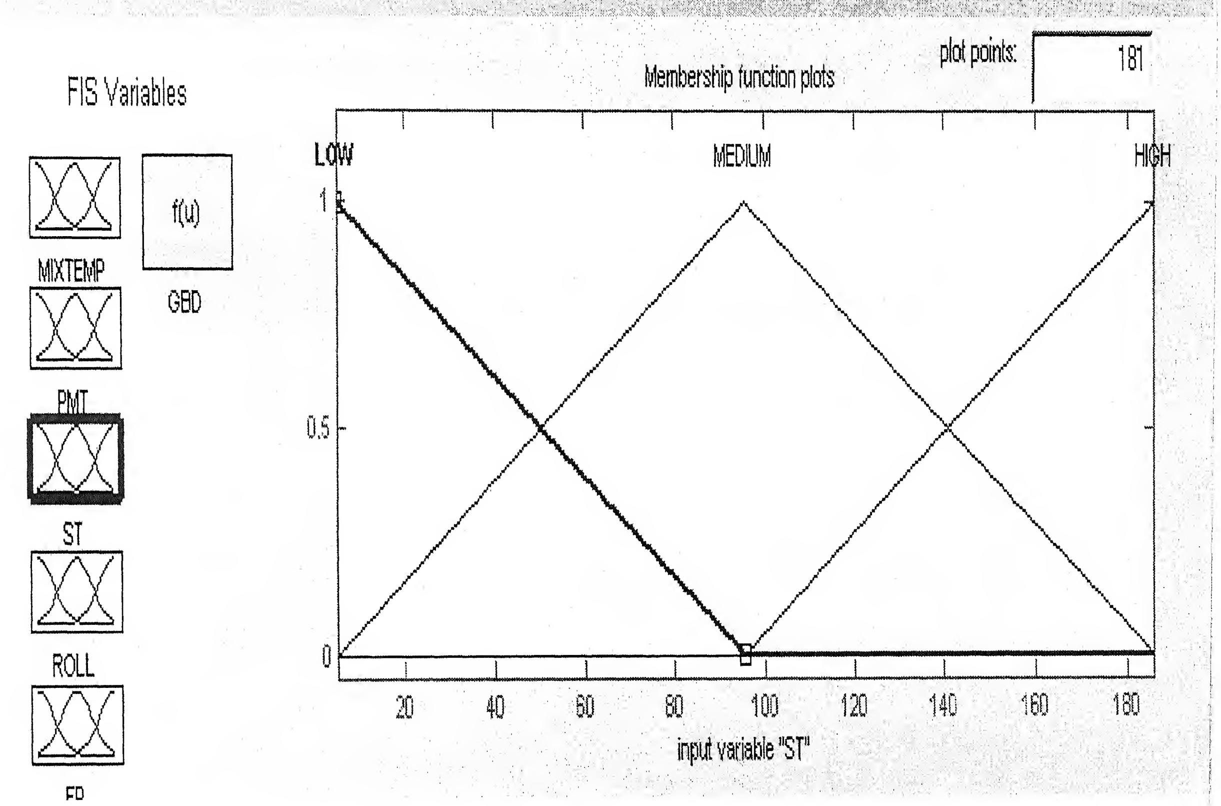
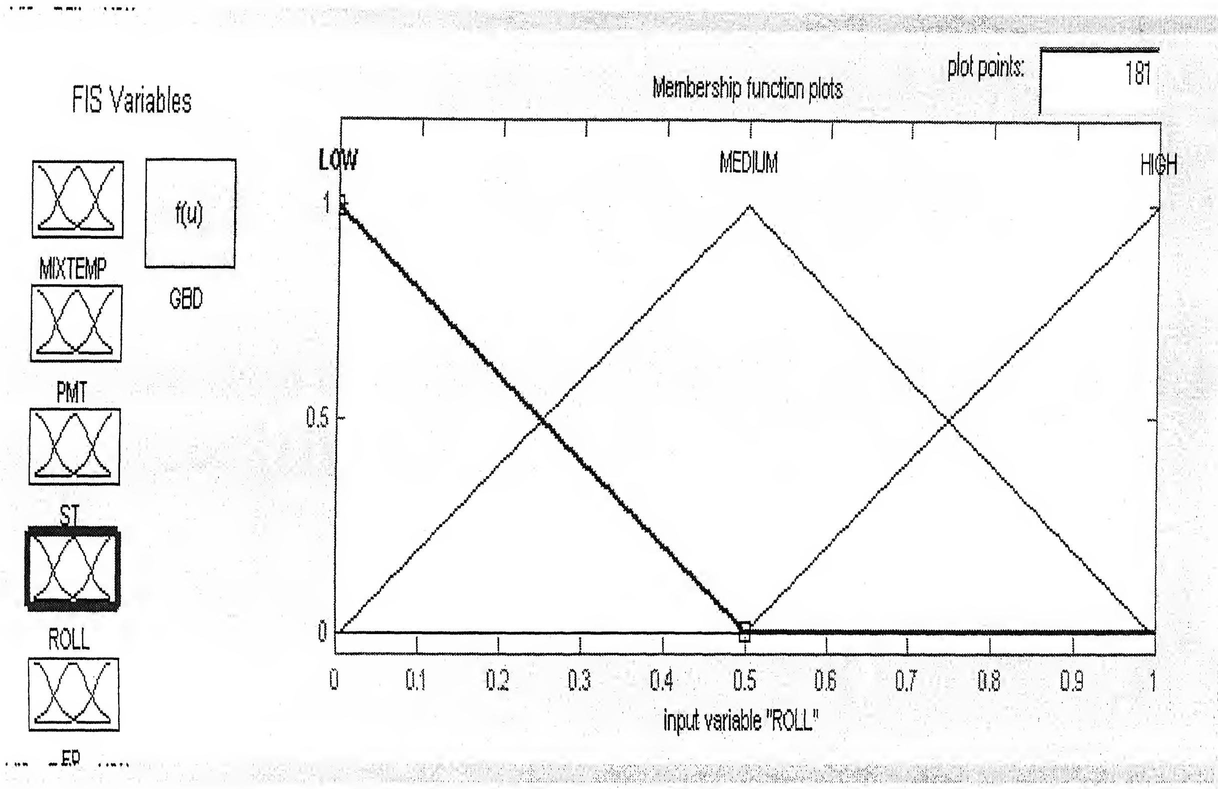


Fig 8.23 & Fig 8.24 Membership Function for PMT & ST



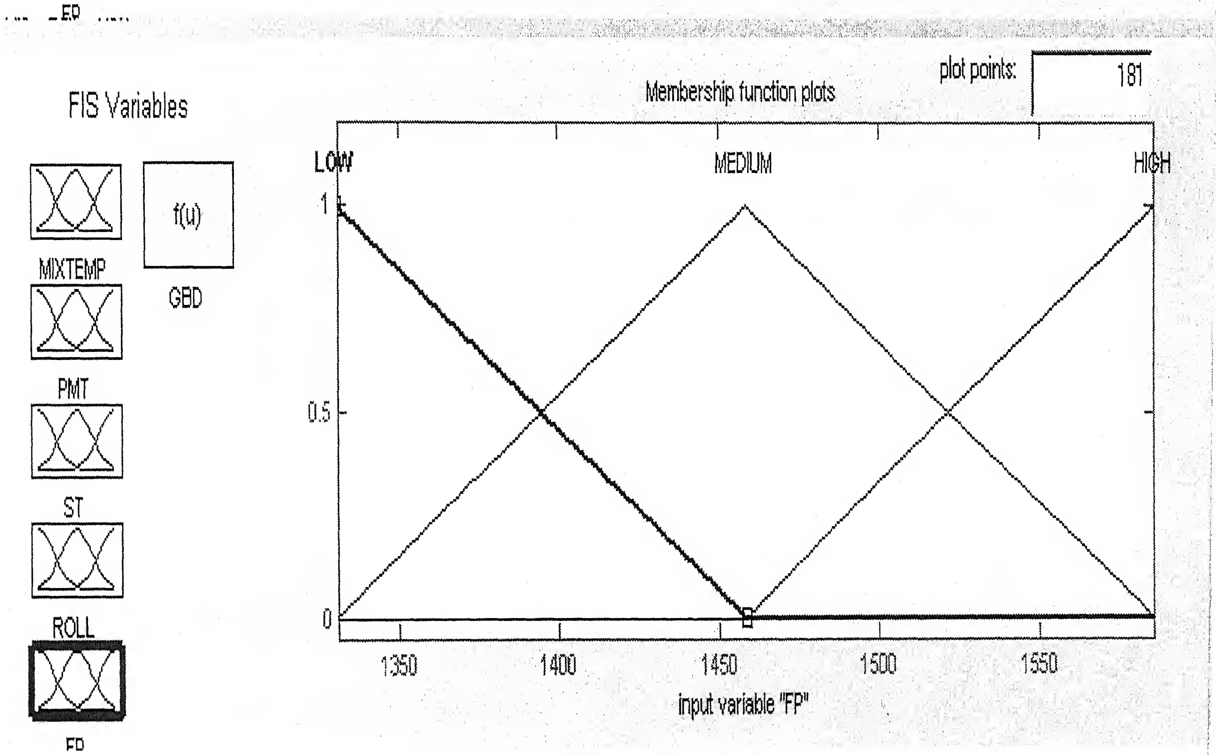


Fig 8.25 & Fig 8.26 Membership Function for No of Rolls & FP

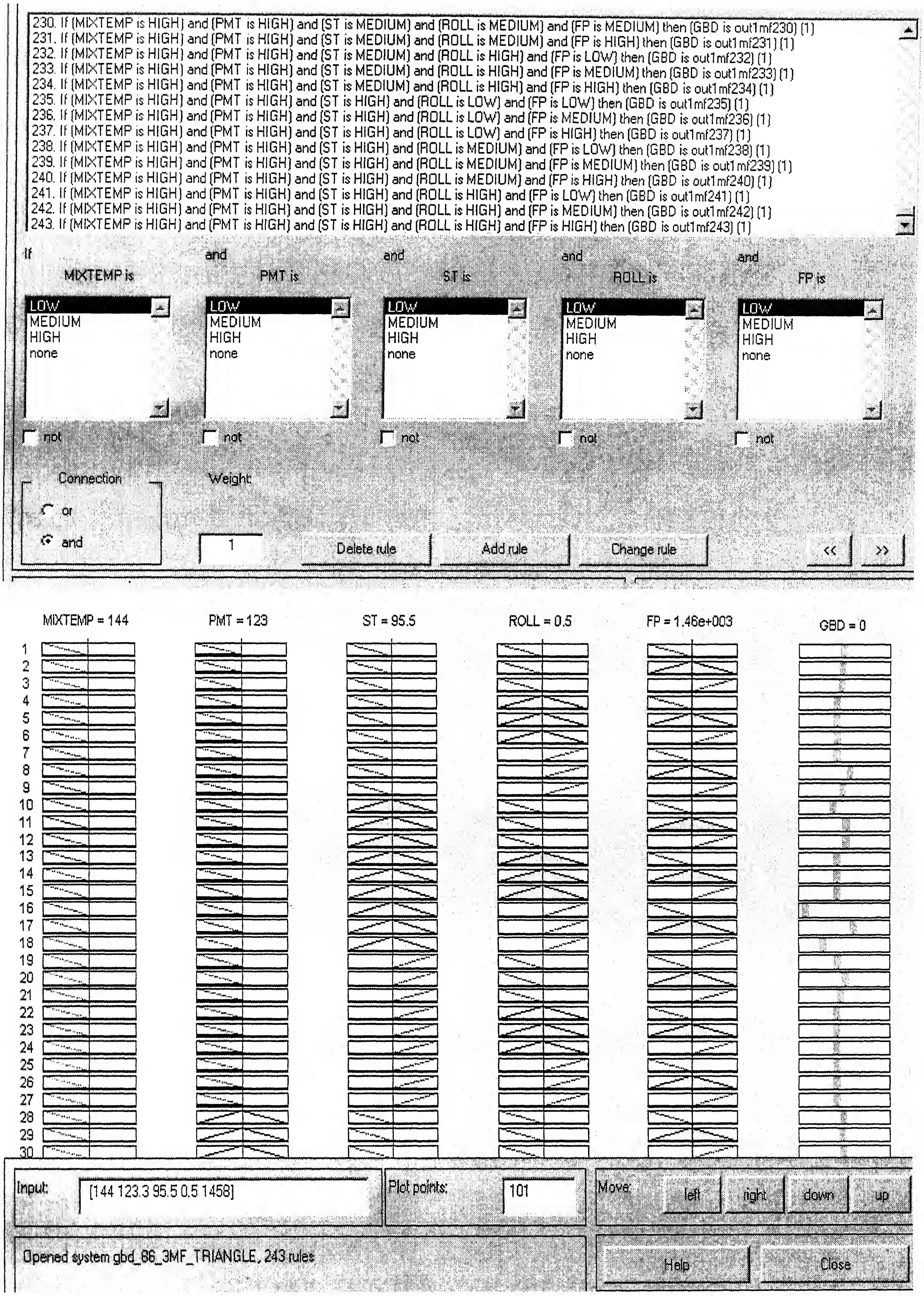


Fig 8.27 & Fig 8.28 Rule Editor & Rule Viewer Consisting of 243 Rules

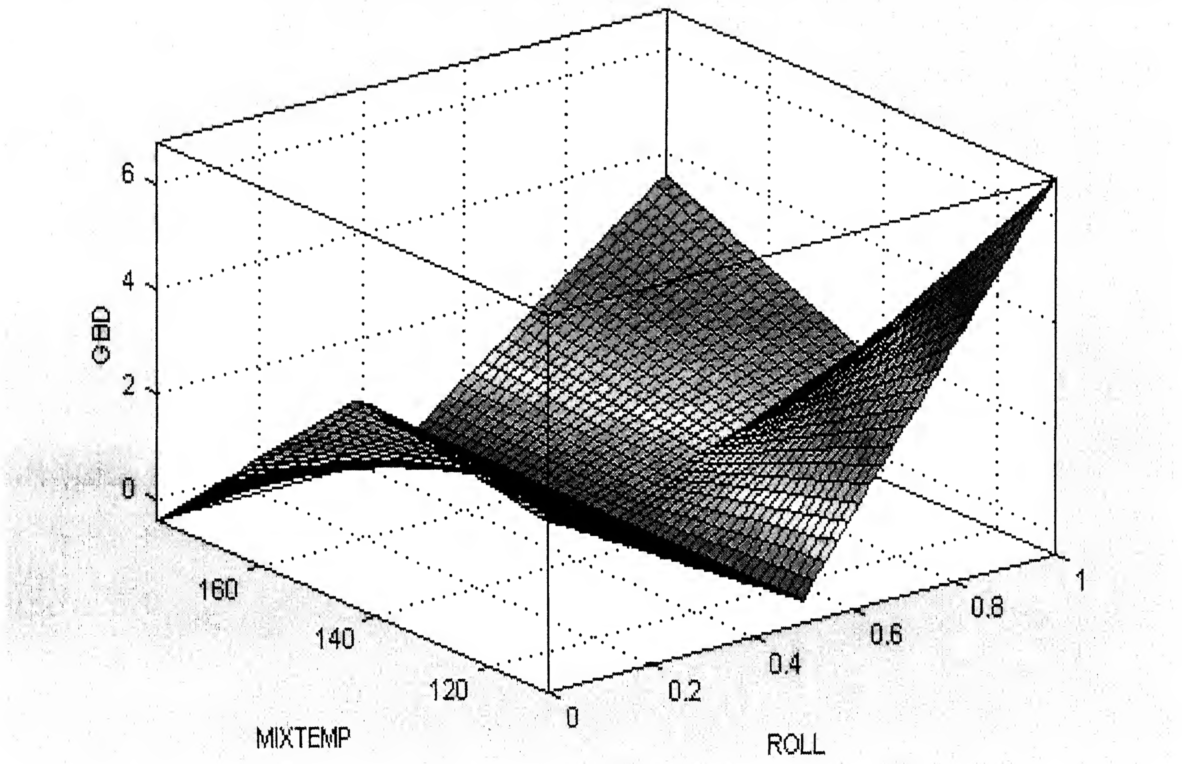


Fig 8.29 Surface Plot for Rolls & Mix Temperature Vs GBD

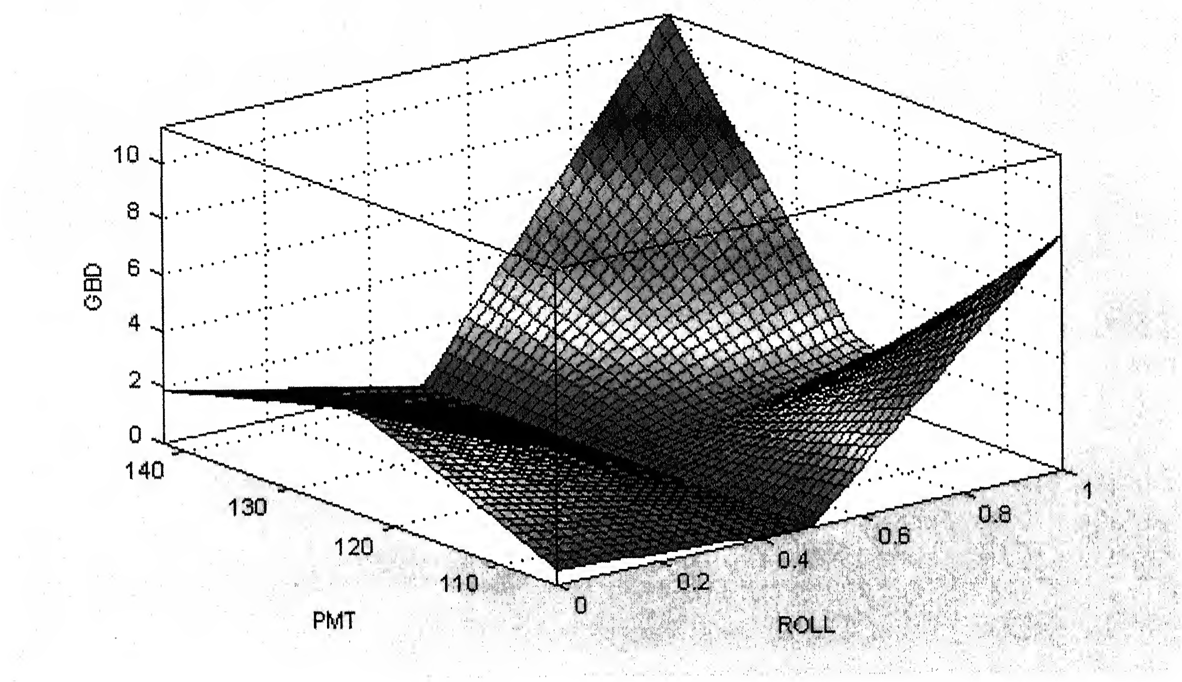


Fig 8.30 Surface Plot for Rolls & PMT Vs GBD

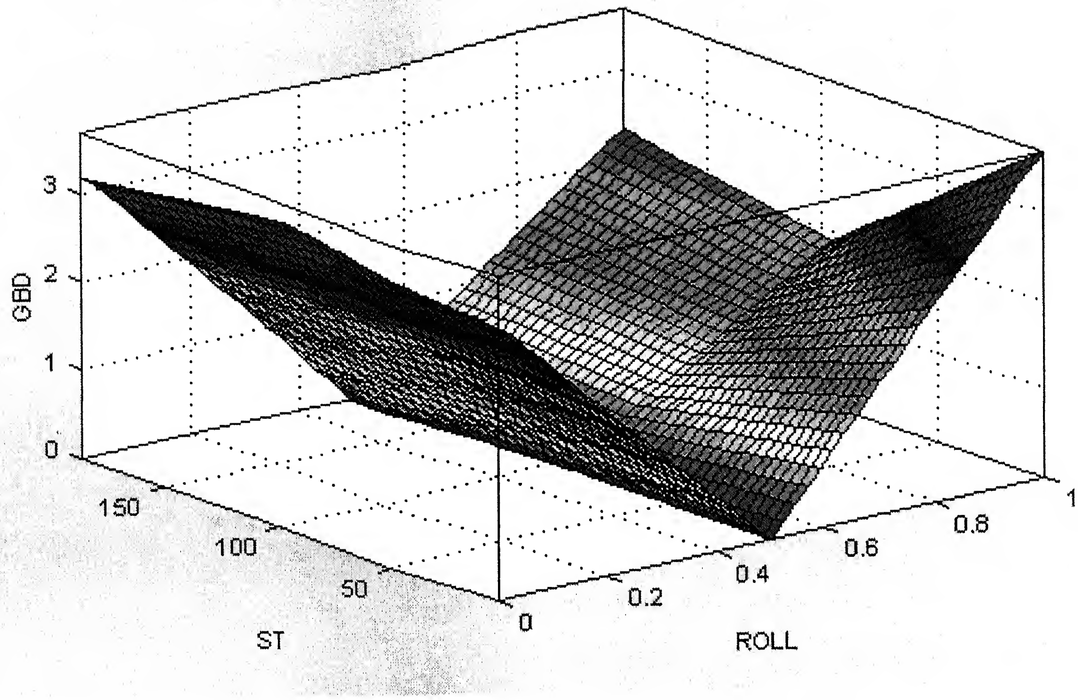


Fig 8.31 Surface Plot for Rolls & ST Vs GBD

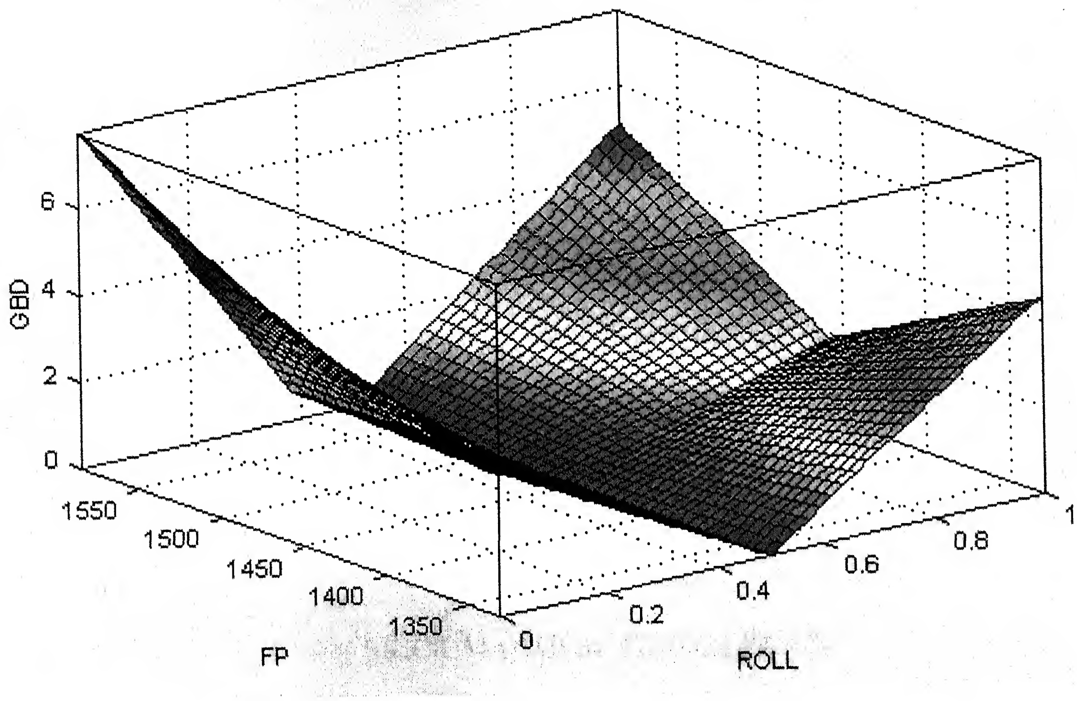


Fig 8.32 Surface Plot for Rolls & FP Vs GBD

REFERENCES

[1] Fuzzy Logic Toolbox, For Use with Matlab, " User's Guide" Version 2, The MathWorks, Inc.

[2] Adrian Biran, Moshe Breiner, " MATLAB for Engineers", ADDISON-WESLEY PUBLISHING COMPANY.

[3] JUN YAN, MICHAEL RYAN, JAMES POWER, " Using fuzzy logic" *Towards intelligent systems,* PRETICE HALL

[4] Project Report, Steel Authority of India Limited, "Introduction and Stabilization of Magnesia Carbon Refractories Production Technology at LDBP of RSP", March 1999.

[5] Timothy J. Ross, " Fuzzy Logic with Engineering Applications", McGraw Hill Inc.

[6] Quinlan, J.R, " Decision trees and decision-making", IEEE Transactions on Systems, Man and Cybernetics, Volume: 20, Issue2, March-April 1990 (pp. 339 –346)

[7] Fabio Gagliardi Cozman, 1998 – 2001, "Javabayes –version o.346, Bayesion networks in Java, User Manual" . Internet link is fgcozman@usp.br, http://www.cs.cmu.edu/~fgcozman/home.html Escola Politécnica, University of São Paulo.

[8] CART for windows Version 4, " User's Guide', Salford systems 1999-2000 , internet link www.salford-systems.com.

[9] Riza C. Berkan, Sheldon L. Trubatch, "Fuzzy Systems Design Principles: Building Fuzzy IF-THEN Rule Bases", IEEE Press, Standard Publishers Distributors, Delhi

Sqn Ldr P A Patil, " EXPERT SYSTEM DEVELOPMENT FOR REFRACTORY BRICK MANUFACTURING PROCESS", Mtech thesis write up 2001-2002.

Date Slip

This book is to be returned on the date last stamped.
